DRAGON USER

The independent Dragon magazine

April 1988

Contents

STOP PRESS: *OS-9 User Group organiser Martyn Vernon has suddenly ceased contact. They are naturally worried. Any information, please send an SAE to Gordon Twist, 88 Stone bridge Drive, East Leak, Loughborough, Leics LE12 6JP.*

Editorial

GLANCING at last month's editorial, I observe nostalgically that the spaghetti overdose is now several weeks behind me. I am otherwise well, thank you. If you were wondering where all the snow was this winter, it's presently lying ten foot deep on the southern alps, but is expected here shortly, I understand.

Back to reality. I hope you all have your Marches. Sorry we're late. The issue was actually ready a bare six days later than usual, but too late for the printers to alter their schedules for us yet again. It arrives as I write this. Bob Harris would like you to know that the KLIK utility costs only £14.95, and not £14.14.95 as stated, and I will adjure you again not to forget the Ossett Show on April 30th (enquiries to John and Helen Penn on 04203 5970), the classic of the Dragon year to many Dragoners.

This month we have a long CAD program for engineering hobbyists — in reply to the constant moan that CAD programs usually cost a fortune and need an Apple Mac, and reviews of two new games which have caused a fair bit of excitement already.

Telephone number
(All departments)
437-4343

Editor
HELEN ARMSTRONG

Production Editor
DRAGON EDITORIAL

Administration
CAROL FRITH

Advertisement Manager
DRAGON EDITORIAL

Marketing Manager
HELEN PERRY

Managing Editor
PETER KANE

Publishing Director
JENNY IRELAND

Subscriptions
UK £14 for 12 issues
Overseas (surface) £20 for 12 issues
ISSN 0265-177. Telex: 296275
Dragon User, 12/13 Little Newport Street, London WC2H 7PP
US address: c/o Business Press International, 205 East 42nd St, New York, NY 10017
Published by Sunshine Publications 1988

Typesetting and Production by Artset Limited, London NW1.
Printed by Headley Brothers Ltd. Ashford, Kent
Registered at the Post Office as a newspaper.

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extent depend on the quality of the discoveries that you can make with your Dragon. The Dragon computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped addressed envelope.

Letters

This is your chance to air your views — send your tips, compliments and complaints to Letters Page, *Dragon User*, 12-13 Little Newport Street, London WC2H 7PP.

Bad news letter

PLEASE could you help me over a problem I am having sorting out. A while ago I read a newsletter which was given away at the Ossett Show, where I went to a Dragon show with the club I am in.

While looking round the show, I stopped at a stall giving away newsletters called *News from the Dragon*, edited by A. Read and others.

After reading the mag I wanted to read more of the mags like this, so I did as requested and sent stamps off to the value of the postage for each mag I wanted. I sent off money for about four mags but I only received one other mag, which to me is a pity because it had helpful hints on programs and games.

I have sent off a couple of letters but have had no reply. Can any readers help me over this?

Dennis Gates
194 Bek Road
Newton Hall Estate
Durham DH1 5LH

HOME-produced magazines come and go at a rate which makes even trade computer magazines look stable. If anyone connected with the erstwhile *News from the Dragon* is reading this, please get in touch with Mr. Gates.

Small magazines usually fold due to exhaustion, poverty and over-optimism on the part of the proprietors, rather than any plot to deprive people of their money and stamps. I would advise readers never to send away for more than one magazine at a time, and not to take out subscriptions until you know the magazine has a track record. The only subs *Dragon User* recommends are *Dragon Update*, *68 Microcosm* and the OS-9 Users Group; and I personally never advise anyone to expect miracles.

Programmer wanted

I AM a comparatively new Dragon user. I am hoping to use my Dragon 32 to assist me in my pricings at work, but I am not proficient enough to write my own programs.

Do you know of anyone who could possibly write a program for me (for a financial consideration, of course)?

Mark Matthews
14 Grantley Close
Ashford, Kent TN23 1UE

Some of the DU readers are very good programmers and I hope someone who feels able to write (or sell you) a watertight costing program for the '32 will get in touch. Have a word with Bob Harris and one or two other Dragon suppliers as well.

Every month we will be shelling out a game or two, courtesy of our supplies, to the reader/s who send the most interesting or entertaining letters. So send us your hints and your opinions, send us your hi-scores and suggestions. Send us your best Dragon stories. What d'you think we are, mind readers?!

EXTRA PUFF

The answer to the question of life . . .

I HAVE owned a Dragon 32 for five years. I also have a Dragon 32 with 64K memory (loaned to me by a friend who seems to have about five). There seems to be a lot of controversy about several subjects and, as a Dragon enthusiast with a wide knowledge of things computerish, I will try to clarify the problems.

The complaints seem to be mostly about 1) power supplies 2) software support 3) 64K games 4) imports.

One at a time, then.

Power supplies: In all the 5 years of Dragon ownership my machine has NEVER gone down. The power supply, on/off switch, etc. all work perfectly. Apparently many transformers do break down, and the replacements are not cheap. However, if like me you own a Dragon 32 with 64K memory, the power supply *is* inadequate. This leads to the voltage regulators overheating. This leads to the adjacent video monitor getting warm. This gives unwanted loss of colour and even picture (many people complain of no colour). Also, this creates a loud hum on the audio circuits which is very annoying when using music. (*Anyone encountering these problems should phone Harry Whitehouse on 0636 705230, since he is the power supply guru these days. He may recommend that you buy one of his PSUs, but he won't if he thinks it won't do the job. — Editor*)

Software support: RIP Microdeal. Quickbeam will go under if they continue to charge high prices. Preston software will (hopefully) do well. In short, software will be written by small companies who are mostly exclusive to the Dragon.

The 64K factor is sad. Microdeal refused to fund *Usurper of Rome* after it became too long for a Dragon 32. According to them, there is little enough of a Dragon market without halving it. But this attitude is taken by all the major software houses.

Imports: We all read about the £40 per game shocker. Microdeal imported *Shocktrooper*, converted it and sold it. Games such as *Paper Round*, *Sailor*, *Brewmaster*, *Gantelet*, *Zaxxon*, *Pooyan*, etc. etc. which are freely available in America are not freely available here. Microdeal don't import them because they run mainly on Dragon 64s. The problem is not conversion (most games run perfectly or with small alterations) but import tax, licensing etc. Only a large company can afford all this. Broomsoft would, but we need funds. There is some excellent software in America, but it would cost a bomb to sell it. Even if we imported and licensed it, we would have to sell at about £10 a shot to recover our losses. I'm sure many owners will write and say, "I don't mind paying £10", but we'd need advance orders and consumers might be disappointed.

Best wishes for the Dragon. I will be happy to answer individual readers' queries on the above subjects.

Michael Edwards
30 Broomhills
Welwyn Garden City
Herts AL7 1RF

THE truth is that people who run substantial software houses are entrepreneurs who are trying to make a living, even if they started for love. Every time they put administrative effort into a shrinking market rather than an expanding one, they are effctively subsidising it out of their own pockets. For this reason, many companies pull out of small markets while the market is still showing signs of health.

I don't think it's fair to single out Quickbeam, as people sometimes do, when many original games were selling for more than half their prices at a time when the market was much bigger. Pam D'Arcy's recent experience has also been that developing games to a commercially competitive standard is just not cost-effective. Put it another way: the cheap original game software now available is being subsidised by the author's free time, full time job and/or family. Utilities are a slightly different kettle of fish, but the same underlying principles apply.

As you point out, most of Microdeal's games were conversions from other sources, so that development costs had already been subsidised before the game ever hit the Dragon market. Your estimate of the approximate selling cost of a licensed conversion in the Dragon market is some indication of how much money is involved.

The same old bug

I enjoy reading every *Dragon User* from beginning to end and always find something interesting. My Dragon 64 with two disc drives (and Superdos) has been harnessed to breathe some fire into my daily activities such as writing letters, invoices, accounts, doing mathematics, a bit of CAD, etc., most of which are my own programs.

There is one bug, however, which DU has in common with pretty well all other electronics and computer magazines, which is that a published program very rarely works first time. The reason is usually a small innocuous error somewhere along the line.

An excellent example of this has crept into the February issue on page 26, the long multiplication. Line 130 should read: 130 Y$=STR$(Z):Y$=MID$(Y$,2). Only then the program works. And I must confess, it took me several evenings until 31st Jan, when I had a sudden flash of inspiration. I think the reason was that on running the incorrect program halted with LS error in 140, not 130. It just shows that even a short program is not as easy as all that to understand fully.

J. Segenschmid
Hill Cottage
Plomer Hill
High Wycombe
Bucks HP13 5NB

American 'color's

I might be able to shed light on a couple of problems. Firstly, Peter Whitaker's word processor was designed to run on DragonDOS. It will not save on CumanaDOS. I once wrote to Peter on this subject and what follows is his advice:

Use the tape version of the program. To save, first exit to Basic, then save the whole program to disc using SAVE "filename",3072,PEEK(4116)*256+PEEK(4117),4151. This saves the whole program complete with text. When you reload the program, and EXEC(4151) it, the text will load with it. Peter's address was then 73 Norwich Street, Cambridge. (*It appears that Peter has finally moved away from that area, and our attempts to find him have proved futile.*)

The second point relates to monitors. I have acquired a b/w monitor with poor definition. When using the text screen or Hi-res green screen I am better off with a TV. However, when you use the black and white display (as with Basic 42 and available on EDIT+) definition is pencil sharp and shimmer free. It is possible that the VDG (being made for the American market) puts out a red/yellow/blue signal as opposed to the red/green/blue signal that we use. This would also explain the lack of an RGB output. There was a company called Rapidvac in Hull who advertised that they would send details on how to convert your display to black and white for one pound. All they did for me was convert my pound into pints, and I heard no more.

I hope this helps. Now all I need is a poke to make *Electronic Author* run on a black and white display.

Ken G. Smith
33 Glack Road
Deal, Kent CT14 9ND

MY unofficial technical department says that, while it is theoretically possible to create full colour additively from magenta/cyan/yellow light sources (these are the secondary colours used subtractively in full colour printing), in practise the primary additive colours, red, green and blue, are used universally in television, and are not affected by the different colour transmission standards used in different countries. (British PAL is in fact an upgrade on American NTSC). Put another way, if the VDG has a separated output available, it will be an RGB output.

Unless, of course, the whole system works like the tourist Twoflower's camera in Terry Pratchett's The Colour of Magic.

In all probability the Dragon is supplying the b/w monitor with an unbalanced colour output. The colours have to be fed in in different proportions to give a uniformly bright display in black and white, otherwise, the definition will be fuzzy.

However, I shall have to leave it to somebody else to reveal the solution.

Crossword

The fifth Dragon Crossword raises its dinky head with conundrums (not for dumdums) from the glorious history of Dragon games. And we have the results from crossword three: the fortunate few, picked up off the mat, were Paul Priestland of Lechlade, who sent us a shopping list a mile long, and Richard Moss of Blackpool, who doesn't want a platform game.

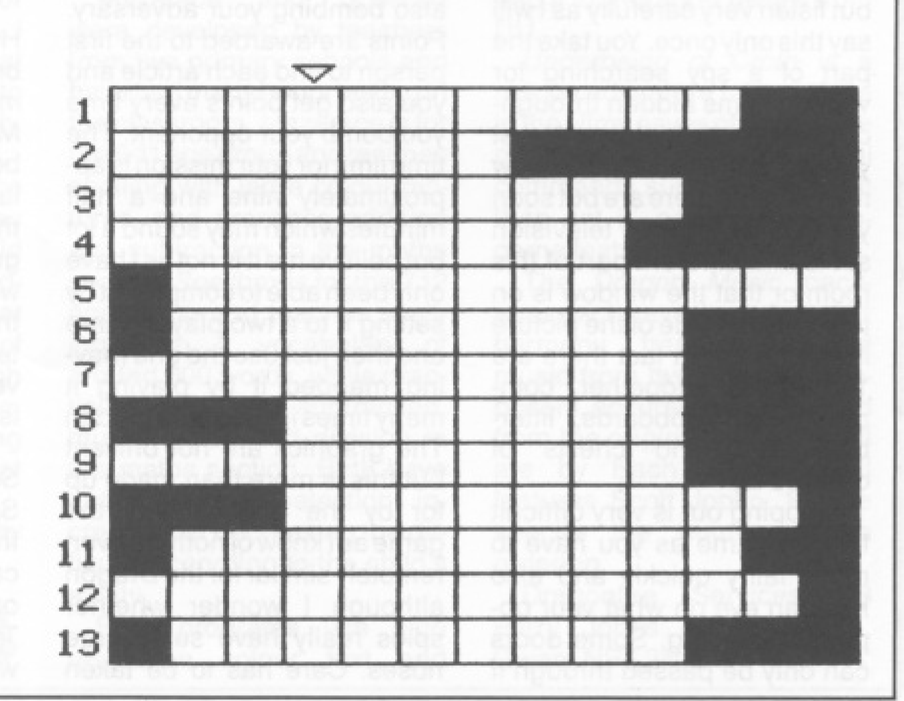
There will be a couple of free tapes from the Editor's Magic Bottomless Box for the first correct entries to reach us each month. You can even try telling us which tapes you'd like in an ideal world. It all depends on what we can find.

And you don't have to cut up your *Dragon User*, either — entries can be written out on a photostat or a plain piece of paper, as long as we can read them.



All this month's answers are names of Dragon software. When the crossword is complete, the column marked with an arrow will spell out a phrase.

1. Steal a chopper from unusual North coast patch (5,6)
2. The fellow with the money? (7)
3. Gripes like anything about German war game (10)
4. Scribbling insect, or a V1 flyer (6,3)
5. Barry rubs gem around to get silly junk food! (5,7)
6. A mere bird to steer a missile by? (4,5)
7. Waifs grown up to make battles fly away! (5,2,3)
8. Sounds like a walking defect from ancient Greece (7)
9. 9. No hoper with twisted mop — the cow jumped over it! (4,6)
10. A doom quiz about a famous campanologist (9)
11. Handle rock, to shut once will let you in on it (10)
12. Fearless Russian who reads hot rod bibles! (5,3,4)
13. Sorcerers in conflict? (6,3)



New software for review should be sent to *Dragon User*, 12-13 Little Newport Street, London WC2H 7PP.

Two is company

Program: Spy Against Spy
Supplier: Pulser
Price: £5.45 (disc)

SHORTLY before Christmas I was in the usual turmoil of working out what extra little presents to buy my eldest son aged fourteen when I heard whispers about a completely different type of game that could be played between two players. I did not even know the name of the game but was told that Pulser Software were selling it so I contacted Brian O'Connor. He is used to me approaching him at the Northern Dragon Shows trying to scrounge software for reviews in *Update* but soon became more affable when he realised that I was ringing up as a paying customer. At first he said that they had no new games software but then said that it must have been '*Spy Against Spy*' that I had heard about. When I explained that it was to be a present he agreed to let me buy a copy as it was playable although not quite ready for marketing. He also promised me a completed version.

It arrived and having booted the disk I was greeted with the usual high standard of Cartwright musical accompaniment and pressed the left fire button to access the one-player option, playing against the computer. I will first explain the general idea of the game but listen very carefully as I will say this only once. You take the part of a spy searching for various items hidden throughout a very large building. At first you do not realise quite how many rooms there are but soon you realise that the television set is in a different part of this room or that the window is on the opposite side of the picture in that room. In fact there are 100 rooms altogether, complete with cupboards, litter-baskets(?), and chests of drawers as.

Mapping out is very difficult for this game as you have to move fairly quickly and also keep an eye on what your opponent is doing. Some doors can only be passed through if you have found the key. The key and the other items cannot be seen but are obtained by moving to the pieces of furniture etc and pushing the Joystick forward, the item being taken if it is there. The screen is split into two with the computer or the second player using the top half of the screen. At first some rooms appear to have no exit until you notice a slight tinge in the middle of the bottom portion of your part of the screen and you realise you can come back out of the room. The first player to find all the items, then escapes by going out through a certain door to be whisked away safely in a helicopter. However you are allowed a limited number of bombs and water traps with which you can take the items off your opponent.

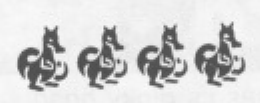
To set these bombs and traps you must either push the joystick forward or pull it back depending on which type you wish to use and then press fire but care must be taken with the positioning of the joystick while doing it as you cannot do this if it is not positioned precisely. Also if the second player discovers an item that the other has already found then he takes it and the first player loses it. The first verion that I played with had unlimited bombs and soon ended up in a mass bombing session with nobody getting anywhere but now one has to plan where to plant a bomb especially as only one will explode if you should inadvertently place more than one in the same room at once.

Scoring is based on discovering the items first and also bombing your adversary. Points are awarded to the first person to find each article and you also get points every time you bomb your opponent. The time limit for your mission is approximately nine and a half minutes which may sound a lot but believe me it is not as I have only been able to complete it by setting it to a two player game and then just use the one (having mapped it by playing it many times in the same mode.) The graphics are not brilliant but this is more than made up for by the originality of the game as I know of nothing even remotely similar for the Dragon although I wonder whether spies really have such large noses. Care has to be taken when moving forward or backwards through rooms as it is easy to move too far and miss a room.

At times I found it difficult to pick up the hidden objects even though I know where they are. I will give the author the benefit of the doubt that this is to simulate searching through drawers. One touch I did appreciate was that if you found the key and entered one of the locked doors you were not prevented from retracing your footsteps if the other player got the key off you.

This shows the thought that has gone into the game as it would have been completely ruined if one player had been trapped behind locked doors. To summarise, this is a good two-player game, which is a rarity in itself, but playing against the computer is not as realistic unless foreign spies are being recruited because they are not very intelligent. Worth it though for the two-player version and nice to be able to practise by oneself.

Mike Stott



A real racer

Program: *Formula One* (Tape or DragDOS disc)
Price: £8.95
Supplier: Pamcomms, John Penn

HOT on the trails of the old and brill *Speed Racer, comes Formula One* by our dear Pam. Many of you out there will have bought *Speed Racer* from the late Microdeal, and will have thought that the super smooth graphics and scrolling track were superb, but let me tell you that *Formula One* is even better. This review is for the tape version, but I think that the disc is identical.

The game is identical to *Scaletrix* for the (dare I say it) Spectrum. It is a split level race the other on the bottom. You can either play against a friend or the computer. There is also a Track Designer on side B, which is also in *Scaletrix*.

The game is in machine code and so is loaded by Cloadm. After loading, you can transfer to DeltaDos disc (instructions on how to do so are given in the manual), or type EXEC to start the game.

First of all, the computer asks you to enter player 1's name, and then player 2's name. If nothing is typed in and ENTER is pressed for player 2's name, then the word Micro appears in player 2's box, and player 1 will play the computer.

Using the right joystick, a little arrow is moved to a selection of boxes marked YES and NO, to select various options.

The first option is to select a track already included in the game. Selecting Yes shows you the 17 different tracks available to choose from. Selecting the icon YES takes you to a question marked 'LAPS ?', where you type in the number of laps you want to make around the circuit from 1-99. If the option to play the computer was chosen, then a message of which difficulty level the micro should have appears. These are 1 Easy (easy to lose), 2 medium (a fair chance), and 3 difficult (very easy for you to lose). After your choice, you go onto the game.

Selecting NO from the 'Use a computer track' option, takes you onto the 'Load a track option', where you can load a track which you have designed using the track designer program. Selecting YES loads the track, and selecting NO takes you back to the 'Use a computer track' option.

The screen display. the screen is split into 3 thirds. Player 1 in the top third, player 2 in the middle third, and the players maps (to show were they are on the track), and the mileometers in the bottom third. The background for the track (the buildings and mountains etc) remain the same for each track, and they turn in the opposite direction to which the car is turning, to make the feeling of movement. To give the feeling of travelling forward, various objects come towards you at the speed of the car. The objects are Cones, Barrels, and Sign posts.

continued on next page

Discs of Destiny

DISKBASE for Dragon DOS (the disc version of *Magbase*, £2.99 all inclusive) and Johnathan Cartwright's adventure trio *Starship Destiny*, *Dungeon Destiny* and *Wild West Destiny* are available in tape and disc versions from Pulser Software for £3.99 each or £8.99 together. Look out for reviews of the trio in due course.

Pulser Software, 36 Foxhill, High Crompton, Shaw, Oldham OL2 7NQ. Tel. 0706 849189.

from previous page

You use the joysticks to control the cars, and the way you use the joysticks is: Forward/accelerate, back/brake, left and right/turn. It is advisable to use potentiometer joysticks where ever possible.

Player one is situated at the top half of the screen, while player two is situated at the bottom half. After a couple of seconds, the lights at the right hand side of each half turn to stripes, and when the joystick/joysticks (Depending on wether one or two players) are pulled back, the lights go white and away you can go. Your top speed is 240mph. The screen scrolls beautifully, and after every lap your time for that lap is shown. Trying to take over your opposition's car is very difficult, and if the two cards collide then it is considered that the person behind is the looser. Pressing BREAK during the game resets the program to the very beginning, and pressing CLEAR takes you back to the 'Use a computer track' option. After the desired number of laps or a crash, then the victory and looser screen is displayed. This shows both the winner and the loser at the same time and the cars are at the same places, one at the top and the other at the bottom. The victor is shown standing beside his car jumping up and down with a victory message underneath, and the loser is shown next to his car stamping his right foot up and down and shouting an exclamation mark to his left.

After the victory/loser screen, the option 'Same again' appears. Selecting YES will take you back to the start line with the same optins selected like no. laps, and Selecting NO, wipes out your loaded in track if any, and takes you back to the 'Use a computer track option'.

The track designer program is on side B of the cassette, and is loaded with CLOADM EXEC. Upon running the prog, the option 'Design a circuit' is displayed. Selecting YES takes you to the designer, while NO takes you to 'Use a computer track' option. Selecting YES lets you add further pieces to the 17 built in tracks, while selecting NO lets you load in a track to finish it off or to check that a track loads alright.

The designer has 15 pieces of track to choose from and four other options. These options are SAVE — save the track to tape, CLOSE — close links the start and end pieces of track together if possible, LIFT removes the last piece of track, and ABORT takes you to the start of the designer program. The 15 pieces of track are Straight A, Straight B, Straight C, Straight D, Standard curve, Standard curve A, Outer curve, Outer curve A, Double Inner curve, Banked curve, Skid Chicane, Chicane Out, Straight chicane, Curved chicane, and Chicane IN. If you pick a curve, then an option to make it go anticlockwise or clockwise is given, OOPS, I almost forgot to tell you that you that there's an option to make you travel north, south, east or west.

Overall, it is a beautiful piece of programming, and it is way better than the three year old *Speed Racer*. I would urge people to buy the game and not pirate it, even at the price of £8.95, because if not enough people buy the game then Pam might have to quit producing software and we don't want that, do we.

David Linsley



Tapez-vous . . . ?

Dragon Taped Computerware tells us that is has now published its *French Collection* (!) including a vocabulary test, a verb test, and a French dictionary. The words for the vocabulary test and dictionary are programmed in by the user with an extensive and easy to use data creation program which is also on the tape. The programs and files can easily be changed for English — Spanish or any number of other useful language combinations.

The vocabulary test was included in the first issue of *Dragon Taped* and the dictionary in the second issue. Both issues are still available at £1.50. Issue two includes English word games, adventure and arcade columns on tape.

People who already have the other two items can buy the verb test for £1. The combined test/dictionary tape costs £3. All orders and enquiries to A R Hopkins, Common Farm, Shifnal, Shropshire TF11 9HN.

Siegfried returns

THE Siegfried Computer-Gruppe of Germany announces a meeting for Dragon enthusiasts to take place on Sunday 26th June 1988 in Jugendzentrum Kratsstation, Honsberger Strasse 2, D-5630 Remscheid, Federal Republic of Germany. Admission 2 DM.

The Gruppe will be demonstrating MIDI, a mouse, sound sampling, a scanner, ram- and eprom-discs, all on the Dragon.

For more information, contact the Siegfried Computer-Gruppe, Dina-Ernstberger-Strasse 23, D-8524 Neunkirchem am Brand 1, FRG.

Fire and games

Dragonfire Services have added to their list of titles *Decathlon* (£3.00), *Dragon Music* (two tapes of Bach, one of Scott Joplin, £3.00 each), the text adventure *Underbeings of Croth* (£4.00) and the spelling/maths tutor programs *Mazerace* and *Spellbase* (£5.00 the pair).

Mazerace and *Spellbox* were developed by teachers from two primary schools and have had many hours testing in the classroom. *Mazerace* is for the 6-11 age group and teaches spelling with large colour pictures, and practices addition and subtraction in the maths section. *Spellbox* is for the 7.12 age group and teaches spelling from a vocabulary of around 600 words, while practicing addition, subtraction, multiplication and division in the maths section. Both have automatic level selection, increasing or decreasing the difficulty according to the child's ability.

Both programs use high resolution colour text and graphics. Previously available from the authors, this package was favourably reviewed by *Dragon User* in 1987.

Decathlon features ten games for 1 to 5 players, including Anagrams, Hangman and Shootout. Ideal for parties and idling away wet afternoons, this package originally had a four-dragon review from DU.

Underbeing of Croth is a rescue adventure taking place in the slimy caves of Croth. The program understands dual commands such as 'pick fruit then eat it', and has a save-game feature.

The *Dragon Music* packages each have three, four part harmony transcriptions of music from two popular composers, accurately transcribed from the originals. DM I and II are by Bach, and DM III features Scott Joplin. Please specify the number when ordering.

Dragonfire Services, 13 Parry Jones Close, Blaina, Gwent NP3 3NH.

Pamcodes

Part five of ***Pam D'Arcy's*** *introduction to machine codes*

REPLACING the subroutine lines UP/MDWN/MLFT/MRGHT with **listing 12** updated following last month's techniques of

a) inserting $ symbols after the # where the required generated code is identical with values in the operands

b) replacing addresses $6400-$6403 in the operand column with position independent WORK0-3,PCR

results in **listing 13** — a fully working, relocatable program on my system.

Did you notice an inconsistency in the listing as taken from the book? The line at address $5069 in **listing 13** actually has a DECIMAL value of 15 in the operand column so I particularly hope DragonDOS users spotted this before experiencing disaster (if typed in as $15=decimal 21, the yellow blob would be allowed to move unseen over the disc workspace area, potentially causing the drive(s) to operate and corrupt any unprotected discs currently loaded in them — I didn't court disaster by experimenting!)

Instruction detail

I will run over the nature of the instructions used and leave you to work out how and why the program works. Basically, unless there are any specific parameters=data specially set up in registers or memory prior to the EXEC or BSR or JSR call, the contents of registers and memory for variables, set aside by the likes of the RMB instruction, are unknown and could contain anything. This is very different from Basic that gives you cleared variables the first time that a variable name is used (so no problems are encountered if the program adds 1 to counters etc. without clearing them first). Clearing or setting up starting values in registers and variables space in assembler is often referred to as *initialising fields*, and a certain amount of this is carried out at the start of the program that we have been working with.

When referring to a location containing zero or being cleared, this is when all its bits are unset (=0 no 1). For clarity, I usually follow it up with the reference to the word null or $00. This is because zero meaning the printable digit 0 has a decimal value of 48 ($30; &H30).

Apart from the special program counter (PC) that is updated internally continuously as the program is running and the condition code (CCR) register that is updated by most instructions as they are executed, values in registers and memory remain unchanged until instructions are executed that amend their content.

Condition codes

Five of the eight bits=flags of the condition code register (CCR) are commonly affected when executing instructions. The conditional branch instructions then allow us to vary program paths according to obtained results to achieve the aims of our program requirements, or specification. Of these five flags, the Half carry and oVerflow flags only tend to be considered for less frequently used types of arithmetic that will not be covered until later in the series. That leaves the common flags Negative, Zero, Carry.

If a resulting condition is true, the flag is set (=appropriate bit is set to 1, $01). Thus if a result is negative, the program will follow a BMI path or not follow the BPL path; a Zero (null) result will follow a BEQ path or not follow BNE path. If the condition is untrue, the flag is cleared or unset (=appropriate bit is set to null) and reverses the above paths.

The carry condition will be dealt with when encountered in a future example. As well as carry resulting from arithmetic instructions, the flag is often used by programmers as a return parameter to a calling subroutine to signify that the likes of validation checks on data passed to it were successful or otherwise. The BCS (Carry Set) and BCC (Carry Clear) are its associated conditional branch instructions.

Other conditional branch conditions, such as BLO and BGE, actually act on individual or combinations of the above flag settings but are readily understandable in the context of source code where they are usually found following CoMPare instructions.

For completeness, all flags affected by the instructions described below will be specified.

Listing 13

```
5001                 * LISTING 13
5001                 *
5001                 * YBLOB13 (FILENAME)
5001                 *
5001                 * THE YELLOW BLOB - PAGE 56
5001                 * FROM "DRAGON MACHINE CODE"
5001                 * BY JONES & COWSILL (SHIVA)
5001                 *
5001                 * CONVERTED TO BE RELOCATABLE
5001                 * (LISTING 11+LISTING12)
5001                 *
5001                 * USING DSKDREAM ASSEMBLER
5001                 * AFTER CLEAR200,&H5000
5001
5001 160004                 LBRA   GO
5004
5004           WORK0 RMB    1         ;FOR 6400
5005           WORK1 RMB    1         ;    6401
5006           WORK2 RMB    1         ;    6402
5007           WORK3 RMB    1         ;    6403
5008
5008 BE0400    GO    LDX    #$0400
500B AF8CF6          STX    WORK0,PCR
500E 108E01FF        LDY    #$01FF
5012 8660            LDA    #$60
5014 6F8CEF          CLR    WORK2,PCR
5017 6F8CED          CLR    WORK3,PCR
501A A780      CLEAR STA    ,X+
501C 313F            LEAY   -1,Y
501E 26FA            BNE    CLEAR
5020 869F            LDA    #$9F
5022 AE8CDF          LDX    WORK0,PCR
5025 A784            STA    ,X
5027 BD8006    KBD   JSR    $8006
502A 27FB            BEQ    KBD
502C 815E            CMPA   #$5E
502E 2604            BNE    DOWN
5030 8D21            BSR    UP
5032 20F3            BRA    KBD
5034 810A      DOWN  CMPA   #$0A
5036 2604            BNE    LEFT
5038 8D2F            BSR    MDWN
503A 20EB            BRA    KBD
503C 8108      LEFT  CMPA   #$08
503E 2604            BNE    RIGHT
5040 8D3D            BSR    MLFT
5042 20E3            BRA    KBD
5044 8109      RIGHT CMPA   #$09
5046 2604            BNE    BREAK
5048 8D4B            BSR    MRGHT
504A 20DB            BRA    KBD
504C 8103      BREAK CMPA   #$03
504E 2702            BEQ    END
5050 20D5            BRA    KBD
5052 39        END   RTS
5053
5053                 * AMENDED LISTING 12
5053
5053 8600      UP    LDA    #$00
5055 A18CAF          CMPA   WORK3,PCR
```

Instruction analysis

LD: LoaD a register. LDA and LDB copies a single byte of data into the specified 8 bit register; LDD, LDX, LDY, LDS, LDU copies two bytes of data into the specified 16 bit or double byte sized register. The data may be an actual value, where the operand is preceded by a # symbol, or be copied from one (8 bit) or two (16 bit) consecutive bytes of memory.

CCR flags: the overflow flag is always cleared=unset. The negative and zero flags are set if the content of the value being copies into the register is either negative or zero (null) respectively; otherwise the flags are cleared=unset. I will definitely go into bytes, double bytes and negative values in the next issue.

Examples from **listing 13** are LDX #$0400 — copies an actual value of $400 (= memory address of the start of the text screen) into register X; LDY #$1FF — copies a count of $1FF (decimal 511) into register Y; LDA #$9F — copies the value of the text screen graphics yellow blob (decimal 159 — Appendix A of the manual supplied with the Dragon computer) into register A; LDX WORK0,PCR — copies the current contents of the two bytes in memory locations WORK0 and WORK1 into register X.

ST: STore contents of a register in memory. STA and STB copies the single byte of data into the specified byte of memory; STD, STX, STY, STS, STU copies the two bytes of data into two consecutive bytes of memory.

CCR flags: the overflow flag is always cleared. The negative and zero flags are set if the content of the value being copied into memory is either negative or zero (null) respectively; otherwise the flags are cleared.

Examples from **listing 13** are: STX WORK0,PCR copies the current contents of register X ($400 as just loaded) into the two consecutive bytes at memory address WORK0 (=locations WORK0 and WORK1); STA ,X copies the value in register A=the just loaded value $9F=test screen yellow blob to the memory address currently contained in register X=current 'cursor' position; STA ,X+ copies the value in register A=the just loaded value $60=text space character when being POKEd (see January 1988 issue) to the memory address currently contained in register X — then does someting else that has not yet been covered! The + is an index mode option known as auto increment, mentioned in the December issue and covered following this section.

CLR: CLeaRs the specified 8 bit operand. Operates on an 8 bit register or byte of memory only; it clears all 8 bits=unsets all 8 bits=makes the byte=$00 (null). CLRA, CLRB or CLR memory are its possible formats.

CCR flags: always clear (unsets) the negative, overflow and carry flags. Always sets the zero flag.

In **listing 13** the memory locations WORK2 and WORK3 are cleared in readiness for arithmetic performed on them later.

LEA: Load Effective Address. As previously mentioned, this is a most powerful instruction in creating position independent code. It can be used with either of the indexable registers, LEAX, LEAY, LEAS, LEAU. However, as in previous examples, in this routine it is seen in its register arithmetic mode.

CCR flags: trust us to start with a special case again! LEAU and LEAS do not affect any flags of the Condition Code Register (CCR). In the circumstance of either of those registers being used for counting down (decrementing), one would need to follow the instruction with one that would indicate when zero had been reached (such as a STore instruction or CMPS/CMPU #0). LEAX and LEAY affect only the zero flag, so that when it is being used as a simple counter as near the beginning of this code, a loop is created by following it with Branch Not Equal (branch if result of the arithmetic not equal to zero).

Examples of LEA in its arithmetic mode in **listing 13** are LEAY -1,Y, subtracting 1 from a count; LEAX -32,X, subtracting 32, the width of a text screen line for effecting the arrow, and adding 32, subtracting 1 and adding 1 to the contents of register X.

BNE: Branch Not Equal. Conditional branch instructions act on the current settings of the CCR only and have no effect on its content. Depending on context, the BNE path is followed if the result of preceding arithmetic is not zero (<>$00), as following the LEAY -1,Y instruction, or the result of a CoMParison of two items is Not Equal, as following CMPA lines further

Listing 14

```
4E21                      * LISTING 14
4E21                      *
4E21                      * DENIS (FILENAME)
4E21                      *
4E21                      * DENIS O'MULLOY'S PRINT NAME
4E21                      * ROUTINE USING DREAM ASSEMBLER
4E21                      * AFTER CLEAR200,20000
4E21
4E20   4E20                       ORG    20000
4E20   BDBA77                     JSR    $BA77
4E23   318C0A                     LEAY   NAME,PCR
4E26   A6A0              LOOP     LDA    ,Y+
4E28   BD800C                     JSR    $800C
4E2B   8159                       CMPA   #'Y
4E2D   26F7                       BNE    LOOP
4E2F   39                         RTS
4E30   44454E4953 NAME            FCC    /DENIS O'MULLOY/
4E3E
```

Listing 13 continued

```
505B  270E              BEQ    ENDUP
505A  8660              LDA    #$60
505C  A7B4              STA    ,X
505E  6ABCA6            DEC    WORK3,PCR
5061  3088E0            LEAX   -32,X
5064  869F              LDA    #$9F
5066  A7B4              STA    ,X
5068  39       ENDUP    RTS
5069
5069  860F     MDWN     LDA    #15
506B  A18C99            CMPA   WORK3,PCR
506E  270E              BEQ    ENDDWN
5070  8660              LDA    #$60
5072  A7B4              STA    ,X
5074  6C8C90            INC    WORK3,PCR
5077  308820            LEAX   32,X
507A  869F              LDA    #$9F
507C  A7B4              STA    ,X
507E  39       ENDDWN   RTS
507F
507F  8600     MLFT     LDA    #$00
5081  A18CB2            CMPA   WORK2,PCR
5084  270E              BEQ    ENDLFT
5086  8660              LDA    #$60
5088  A7B4              STA    ,X
508A  6A8DFF78          DEC    WORK2,PCR
508E  301F              LEAX   -1,X
5090  869F              LDA    #$9F
5092  A7B4              STA    ,X
5094  39       ENDLFT   RTS
5095
5095  861F     MRGHT    LDA    #$1F
5097  A18DFF6B          CMPA   WORK2,PCR
509B  270E              BEQ    ENDRGT
509D  8660              LDA    #$60
509F  A7B4              STA    ,X
50A1  6C8DFF61          INC    WORK2,PCR
50A5  3001              LEAX   1,X
50A7  869F              LDA    #$9F
50A9  A7B4              STA    ,X
50AB  39       ENDRGT   RTS
50AC
50AC
```

Listing 15

```
5001                    * LISTING 15
5001                    *
5001                    * STRINGS (FILENAME)
5001                    *
5001                    * PRINT A STRING ROUTINE
5001                    * USING DSKDREAM ASSEMBLER
5001                    * AFTER CLEAR200,&H5000
5001
5001
5001 BDBA77                    JSR    $BA77
5004 308C19                    LEAX   NAME,PCR
5007 8D0D                      BSR    PRINT
5009 308C6C                    LEAX   NL,PCR
500C 8D08                      BSR    PRINT
500E 308C1E                    LEAX   INFO,PCR
5011 8D03                      BSR    PRINT
5013 8D01                      BSR    PRINT
5015 39                        RTS
5016
5016                    * PRINT SUBROUTINE
5016                    * ENTRY: X=MEMORY ADDRESS OF
5016                    *        FIRST BYTE IN STRING
5016                    * EXIT : NULL BYTE ENCOUNTERED
5016                    *        IN STRING
5016
5016 A680          PRINT  LDA    ,X+
5018 2705                 BEQ    END
501A BD800C               JSR    $800C
501D 20F7                 BRA    PRINT
501F 39            END    RTS
5020
5020 44454E4953    NAME   FCC    /DENIS O'MULLOY/
502E 00                   FCB    0
502F 44454D4F4E    INFO   FCC    /DEMONSTRATION OF/
503F 4F4E45204D           FCC    /ONE METHOD FOR/
504D 0D                   FCB    13
504E 5052494E54           FCC    /PRINTING STRINGS/
505E 5553494E47           FCC    /USING A COMMON/
506C 0D                   FCB    13
506D 535542524F           FCC    /SUBROUTINE/
5077 00                   FCB    0
5078 0D00          NL     FCB    13,0
507A
507A
```

down the code. Its complement is BEQ, Branch on EQual.

JSR: Jump to SubRoutine. In its simplest understanding, Jumps to the instruction at the memory address specified in the operand and returns to the following instruction when its RTS (ReTurn from Subroutine) is encountered. Although I have generally alluded to JSR ringing warning bells of impending position dependent code, it IS possible for it to have an operand of LABEL,PCR which generates position independent=relocatable code (unlike BSR, the ,PCR MUST be present, or else an extended address=position dependent code will be generated). The generated code is a byte longer than its BSR/LBSR conterpart, although there is no need to differentiate between long and short branches. For readability and compactness, BSR is recommended when writing relocatable code.

CCR flags: none affected

BEQ: Branch on EQual. The complement of BNE.

CMP: CoMPare. CMPA, CMPB compares the specific 8 bit register with a single byte of data; CMPD, CMPX, CMPY, CMPS, CMPU compares the specified 16 bit or double byte sized register with two consecutive two bytes of data. The data may be an actual value, where the operand is preceded by a # symbol, or be one (8 bit) or two (16 bit) consecutive bytes of memory. Compare leaves the CCR set for appropriate conditional branching.

CCR flags: negative, zero, overflow and carry flags (plus half carry unpredictably affected with CMPA/CMPB) are ser/ cleared as appropriate. There are, however, specific comparison result branches, eg BLO to BGE, that are more logical to use than trying to operate on the actual flag information. Signed and unsigned branches will be dealt with next month.

Example from **listing 13** are CMPA (keypress) with actual values (ASCII values for arrow keypresses and break key); in the cursor movement routines, the cursor limits (columns 0 and 31; rows or line 0 and 15) are loaded as actual values into register A and are compared with the current contents of WORK2,PCR snf WORK3, PCR (that are the current cursor column and row values) to check that the user is not trying to move the cursort beyond the screen edges.

BSR: Branch to SubRoutine. The usual, logical subroutine branch instruction when writing position independent= relocatable code. In its simplest understanding, branches to the instruction at the memory address evaluated from the (label name) operand and returns to the following instruction when its RTS (ReTurn from Subroutine) is encountered. If branching >127 bytes away, requires the Long branch form of the mnemonic (LBSR). Assemblers usually advise of instances where a long branch is needed rather than any short ones used.

CCR flags: none affected

RTS: ReTurn from Subroutine. In its simplest form, returns to the instruction following the last BSR, LBSR or JSR carried out.

CCR FLAGS: none affected.

DEC: DECrement. Subtracts one from the specified 8 bit operand. Operates on an 8 bit register or byte memory only; DECA, DECB or DEC memory are its possible formats.

CCR flags: negative and zero. The overflow flag is set if the original operand was $80, else is cleared.

In **listing 13**, DEC is used to subtract one from the current cursor position for valid up and left arrow keypresses.

INC: INCrement. Adds one to the specified 8 bit operand. Operates on an 8 bit register or byte memory only; INCA, INCB or INC memory are its possible formats.

CCR flags: negative and zero. The overflow flag is set if the original operand was $7F, else cleared.

In **listing 13**, INC is used to add one to the current cursor position for valid down and right arrow keypresses.

Monthly workout

To conclude our work on YELLOW BLOB — do you think that the program is error free? How long is the text screen? Have you learnt a shorter coding method of clearing the screen than using the CLEAR loop in **listing 13**?

Winners and Losers

Every month Gordon Lee will look at some prize programming

A real toughie November's competition turned out to be — but with the dark winter nights, what better excuse is there than to wrestle with a real brain teaser? Some competitors thought otherwise:

"My hair is falling out quick enough without competitions like these." — Phil Sapiro

"With competitions like these, *Dragon User* will probably lose some friends!!!" — Mark Towlson

Nevertheless, a number of excellent solutions were received. Nearly all competitors found the quick way of calculating the total of any six-letter word once each letter had been converted to its alphabetical rank. This was to multiply each of the letter values by 1, 5, 10, 10, 5 and 1 respectively. So for the word DRAGON we would get:

Alphabetical	position	factor	product
D	4	1	4
R	18	5	90
A	1	10	10
G	7	10	70
O	15	5	75
N	14	1	14
TOTAL		=	263

Before the above routine can be applied, the words to be tested must be selected, and it was here that the competitors showed much variety and ingenuity. Basically, there were three modes of attack:

1) Typing in a list of words, for example, from a crossword compiler, and testing each word in turn to find its total value. Some entries included the list as DATA lines — typically containing six to seven hundred words — while others relied on each word being input in turn before the test was made. Both of these techniques rely on the availability of a crossword compiler to easily select words of six letters, plus the need for a reasonable typing speed to facilitate an easy inputting of the data.

2) Using a 'seed' word and varying individual letters depending on certain parameters. For example, taking the word DRAGON which we know to have a total of 263, any letter can be changed if its complementary letter is also altered by a reverse amount. For instance, if the initial letter, D, is altered to a C, the total can be belanced by changing the final letter to an O. This will produce the sequence 'CRAGOO' which can be accepted or rejected as a possible word. In the same way, the R and the O can be 'balanced' as these letters both have the same multiplication factors, as do the middle pair of letters, A and G. The advantage of this method is that the middle pair of letters can only have a limited range without taking the total above 263. Unfortunately, competitors who used this approach tended to score fewer words than those using other methods, but this is an idea which, one gets a gut feeling, could produce some interesting results if correctly handled. One would have to remember that there would need to be a certain amount of 'cross-exchanges' between the letters if this method was to be fully explored. A typical cross-exchange would be to alter the R of DRAGON to an S and balancing it by, reducing the final N by *five* letters to an I.

3) Generating permutations of letters and scanning visually to check for the presence of any acceptable words. This technique produced the best results, although certain additional techniques are necessary to reduce the number of permutations that would otherwise be possible. Without any restrictions there are almost 308 million permutations of six letters which needed to be whittled down to the two dozen or so words which were finally discovered. This was generally achieved by taking as a starting point only those two-letter combinations which can begin a word. Thereafter, each letter is appended in turn, provided that its value does not cause the final total to be exceeded. This is the technique used by Mark Towlson, who managed to score the highest number of words, despite being sent 'boz-eyed over quite a number of nights'. Mark's listing is given here. The 26 DATA lines are used to denote all possible first and second letter couplings in a rather ingenious way. Each of the lines relates to each initial letter in turn, and the position of the 1s within that line denote the second letter coupling. For example, the 17th DATA line (representing 'Q', the 17th letter), has a 1 only at position 21, indicating that the combination 'QU' is the only one possible.

The last word this month goes to Keith David who tried a number of approaches and concludes his letter: "The final approach is to do it the hard way. Thiat is, to write a program to test inputted six-letter words. The facility was added to store and recall any successful words found. Optimistically, the storage string was dimensioned to hold 200 such words, and I expected to fill this in about half an hour. After many frustrating evenings at the keyboard, I have now found five words, one of which is dubious!"

For Keith, and all other competitors, a full list of acceptable words (from *Chamber's*, *Webster's* or the *OED*) is printed here.

```
Words totalling 263:
ACHING CHEESY COCKET CODIFY COIFFE CYBELE DEARES DRAGON
HOMAGE ITCHES MICKLE MOBILE MUCATE MUDDLE NILGAI PEGLEG
REDEYE RENAME SACRED SAILED SAMIAN SCALPS SCAMPI SEARED
SHAIRN SHIELD SIDERS SOCKED SOGGED SPICES STEAMS SWEDEN
WIFELY
```

```
10 REM DRAGON USER NOV 1987 PUZZLE
20 DIM Z$(26):FOR I=1 TO 26:READ Z$(I):NEXT I
30 DATA 01111111111111110111111111
40 DATA 10001001100100100100100010
50 DATA 10001001100100100100100010
60 DATA 10001001100000100100101010
70 DATA 11111110101111111111111110
80 DATA 10001000100100100100100000
90 DATA 10001001100101100100100010
100 DATA 10001000100000100000100010
110 DATA 11110110000111100111010000
120 DATA 10001000100000100000100000
130 DATA 10001011100101100110101010
140 DATA 10001000100100100000100010
150 DATA 10001001100000110000100010
160 DATA 10001010100000100000100010
170 DATA 11110110100111010111111111
180 DATA 10001001100101100111100010
190 DATA 00000000000000000000100000
200 DATA 10001001100000100000100010
210 DATA 10101001101111111001101010
220 DATA 10001001100000100100101011
230 DATA 00000000000111010110000000
240 DATA 10001000100000100000100010
250 DATA 10001001100000100100100010
260 DATA 10001001100000100000000000
270 DATA 10101000100001100100101000
280 DATA 10001001100000100000100010
290 COUNT=0
300 FOR A=1 TO 26:A1=A:P$=Z$(A)
310 FOR B=1 TO 26
320 IF MID$(P$,B,1)="0" THEN 490
330 B1=B*5
340 FOR C=1 TO 24:C1=C*10
350 Q1=A1+B1+C1: IF Q1>247 THEN 490
360 FOR D=1 TO 24:D1=D*10
370 Q1=A1+B1+C1+D1: IF Q1>257 THEN 480
380 IF Q1<131 THEN 470
390 FOR E=1 TO 26:E1=E*5
400 Q1=A1+B1+C1+D1+E1: IF Q1>262 THEN 470
410 F=263-Q1
420 IF F>26 THEN 460
430 COUNT=COUNT+1
440 IF INKEY$<>"" THEN GOSUB 520
450 PRINT CHR$(A+64);CHR$(B+64);CHR$(C+64);
CHR$(D+64);CHR$(E+64);CHR$(F+64);"  "
;
460 NEXT E
470 NEXT D
480 NEXT C
490 NEXT B
500 NEXT A
510 STOP
520 PRINT USING "####### ";COUNT;
530 IF INKEY$="" THEN 530
540 RETURN
```

Expert's Arcade Arena

Write to 'The Expert' at Dragon User
12-13 Little Newport St
London WC2H 7PP.

HI folks. Once again, it's Arcade Arena time and this is, of course (not that I need to remind you, you're a trusty crowd), the wonderful one year and eleven months birthday issue. So, to celebrate this prestigious and historic occasion, I am pleased to present this rather splendid *Module Man* map for your delectability. (*Here, can we have it in pen next time please, your greatness? Pencil comes off on the readers' fingers.*)

My thanks to Philip Thomas for the map layout and accompanying notes. He must have sacrificed a few days' work to produce it.

1. The objects that can be carried are a yellow key, a blue key, a shield, a ladder and a sword.
2. Doors labelled with numbers can be passed through if you carry one of the above objects.
3. Doors labelled with letters can be passed without difficulty.
4. Once you have walked through any of the numbered or lettered doors, you will find yourself in the corresponding screen with the same lettered or numbered door, eg, should you pass through door 1 in screen A1, you would find yourself by door 1 in screen A4, etc.
5. Doors marked by circles are 'one way' doors and only appear when you pass through them from the other side of the door. For example, door V in screen A2 can only be reached from door V in screen B5.
6. There are three doors marked on the map which are invisible, which are:
   Door L in screen A4
   Door M in screen B4
   Door W in screen B5
   To access these doors you must find the ladder and drop it onto the platform above the door, then climb down the ladder and use the door as normal.
7. On screens C1 and B4, the letter I by the ladder indicates that the ladder is invisible.
8. The skulls which can be found distributed randomly throughout the screens give extra energy, though it is not advisable to use them, unless your energy is low, in case you should need more energy later.

Catacomb Crisis cheat

Load the first program and once it runs, press the RESET button and type "CLOADM" to load the next part.
Once loaded type:
POKE &H5DFD,10
POKE &H5E01,&H7F
POKE &H5E44,10
EXEC &H7608

Thanks again, Phil, and now to *Catacomb Crisis*, which is a game that I haven't got a copy of! So I can't comment on it, or even discover what the cheat does, but many thanks to Darryl Gove.

THERE IS A SUBTLE HINT IN THE ABOVE PARAGRAPH!

Well, that just about wraps up this month's column, except to squash in a *Moon Cresta* cheat, as it seems that it came out gibberrrriiiiishshsh last time. Second time lucky, as they say. I'll be back next month, rain or shine, when, once again, it will be *Airball* month. This is not only because it's an excellent and very popular game, but also due to the fact that every time I mention the game, I receive a dozen extra sacks of mail. Bye for now . . . *Airball, Airball, Airball, Airball, Airb*

Mooncresta cheat

To load type:
SKIPF:POKE 126,6; POKE 127,) :
EXEC 46941

Once loaded, enter the patch below and then "EXEC 9216". This gives immunity to your smallest ship.
24351: 204,1,2,142,96,68,
237,129,231,128,32,6

(ie POKE 24351,204 then
POKE 24352,1 etc.)

MODULE MAN

by Philip Thomas

Circuits on screen

D A Craig presents a program which draws up electronic circuit diagrams for designers

HAVING been interested in electronics for some twenty six years I have built various projects, all of which, of course, started off as a diagram on a page. This program was developed to help produce a reasonable diagram as well as help me learn to program using my Dragon 32. The program just grew as I though of more facilities to add and this is the result.

The cursor is a pixel, which starts off in the centre of the screen; it is moved by the arrow keys. By holding down the shift key and using the arrow keys the cursor jumps in steps of 20 screen points. The comma, full stop, less than and greater than keys move the cursor to the four corners of the screen and by pressing @ the cursor goes to the centre. On pressing L for LINE the cursor move keys now draw solid lines. To return to the cursor press C whereupon the pixel reappears offset slightly from the line. If you need to rub out a line or part of a line press E for ERASE and go over the line again.

There are 50 different symbols stored in 50 strings selected by a letter A to Y in two groups. Also there are six wound components in vertical and horizontal planes in group 3. The size and angle of the symbols in groups 1 and 2 are selected by the numbers 1 to 4 for size and the arrow keys for the angle. To draw a resistor for example you move the cursor to where you want the resistor drawn, press 1 to select the group, press A to select the registor then press 1 to 4 to select size and finally an arrow key to position the resistor in steps of 90 degrees. Should you make a mistake a sound is produced and an error message appears telling you what you did wrong and reverting back to the drawing screen so you can repeat the selection. The wound components are selected by pressing 3 and the keys A to L, again if you make a mistake an error message comes up.

Wipe out

If you find that you have drawn the wrong symbol it can be erased by pressing the * key provided no further movement of the cursor has been made. Otherwise you will have to use the large scale wipeout routine which is selected by first moving the cursor to the top right of the part to be wiped out. Pressing CLEAR and then the down arrow which draws a line down the screen to the bottom of the component you wish to erase, now press the left arrow key until the erasure is complete. The entire screen can be wiped out by pressing the = key and following the appropriate action to the prompt question.

A symbol guide is available by pressing the ? If you have not drawn anything the guide pages for all three groups appears, otherwise if you have selected a group then only that group guide comes up. Groups 1 and 2 have two pages of symbol guide, the second page showing by pressing N, or, if N is not pressed, the program goes back to drawing mode after a delay.

The diagram can be labelled by the text routine in either vertical or horizontal mode by pressing T and then in accordance with the instructions selecting H or V. The cursor disappears in text mode but pressing ; shows it for a few seconds. The space bar gives a space in the text mode. To exit text press ENTER.

After your diagram has been drawn pressing S starts the screen save sequence. Instructions are easy to follow and the screen is recorded as a machine code file onto the cassette.

The tricky bit

Now comes the tricky bit. After the CIRDRAW program there is a screen dump program on the tape and by pressing P another instruction page appears giving you the print diagram procedure. Originally I had a Basic screen dump but this was too slow, taking approximately 20 minutes to print out a screen, also it printed white on black which was hard on printer ribbons and the printout was rather small, so I tried out a program for a machine code dump originally written for a brother 1009 printer and published in the November 1987 *Dragon User*. My printer is the Taxan Kaga 810, which is Epson compatible as is the Brother printer refered to in the original article. This was much better although I could not append the screen dump onto the CIRDRAW program. The solution I came up with was as described in the instruction page which appears on pressing P. The screen dump program is loaded and autorun on pressing C. A last minute option to go back and check or make modifications is gven by pressing R.

The autorun program attached to the screen dump is the one published in the July 1984 *Dragon User* (thank you, Brian Cadge). Now for a brief explanation of how the program itself works.

Lines 1 to 3 are just to facilitate an easy recording routine while the program was being developed, you may recognise them from the book *The Working Dragon*. The lines 10 to 300 print the introduction page, instruction pages and select the diagram load or instruction display routines.

There are various functions available and these are selected in the lines 300 to 750 by the very useful inkey$ function. An autorepeat routine on the arrow keys moves the dot cursor around the screen with ability to jump to five different locations, ie the four corners and the centre by using the full stop, greater than key for bottom and top right and comma and less than key for bottom and top left. The @ key is used to bring the cursor back to the centre. The cursor jumps twenty screen points by using the shift key along with the arrow keys.

Each symbol has a different letter and by using the letters A to Y a draw string table for each symbol, giving 25 symbols, is made up. These tables can be made up into groups to provide as many symbols as you require, bearing in mind of course the memory used. The group 1 symbols are compiled in the lines 750 to 1050.

We must be able to select which symbol to draw so the lines 1050 to 1350 detect which letter is chosen and puts into Z$ the draw string corresponding to the chosen letter. The Z$ is then used as the string to be executed in the final draw command.

Scale and angle of the symbols are decided in the lines 1350 to 1550, along with an appropriate error trap for each function. While the lines 1550 to 1650 make the necessary conversions from numerical values to the string format for the draw function.

Lines 1650 to 2750 are just text pages giving details of what letter is what symbol in group 1, with the draw strings for group 2 and the group 2 letter guide. A useful trick is the combining of a letter detection routine within a delay loop giving the user the ability of calling up the second information page with a single key; if the first page contains the required information the program returns to the drawing mode automatically after the delay.

Next come the wound components, such as coils, chokes, transformers etc. There are six of them drawn with the circle, line and draw commands. Because there are no angle or size commands in the circle or line functions these components are drawn in vertical or horizontal planes selectable by the appropriate letter. Lines 2750 to 4550 cover the selection routine and error trap, the subroutines which draw the components and the wound components letter guide.

Machine code

That is the drawing section of the program complete. We now move on to the cassette save and load routine in lines 4550 to 4750. The saving of the diagram is done by recording as machine code the memory locations which comprise the four pages used in the pmode 4 display. A tape locating routine is provided to prevent accidentally erasing previous diagrams. The diagram you wish to save is given a filename and then the screen informs you that recording is taking place. To load a diagram from tape you select L when you run the program initially, or D if you wish to load a diagram while the program is still in use. The screen is cleared to white and the diagram is drawn as the information comes in from the tape.

One drawback with the Dragon is the inability to produce text on the hi res screen, so lines 4750 to 5900 draw the numerals and alphabet in a similar way to symbols. The number keys with shift I have used to create special characters such as the mili, nano, ohm etc. These are all listed in the text information pages. Some keys do not draw anything but give a sound to indicate a blank string.

If in drawing a diagram you discover that some parts are not correct then lines 5900 to 6050 comprise a routine for drawing a line to the right of the section you wish to redo and then wiping it out. The auto repeat is used on the sown and left arrow keys by peeking the memory which is concerned with these keys. Once the digram has been drawn, labelled and saved the lines 6050 to the end print out the instructions for loading the screen dump program will be deleted but the choice is given to go back and check that everything is as you want it before continuing.

```
1 GOTO 3
2 MOTOR ON:FOR D=1 TO 10000:NEXT:CSAVE"CIRDRAW1"
3 CLEAR 200:X=128:Y=96
20 CLS:PRINT"CIRCUIT DIAGRAM DRAWING PROGRAM"
30 PRINT:PRINT"    IF YOU WISH TO LOAD A          PREVIOUS DIAGRAM CHANGE TO
 DATA CASSETTE AND  PRESS L":PRINT:PRINT"IF YOU REQUIRE THE INSTRUCTION  PAGES D
URING PROGRAM USE PRESS H"
40 PRINT"IF YOU REQUIRE TO LOAD ANOTHER  DIAGRAM WHEN PROGRAM IS IN USE
  PRESS D":PRINT:PRINT"     DO YOU REQUIRE PROGRAM           INSTRUCTIONS (Y/N
)"
50 AN$=INKEY$:IF AN$="" THEN 50
60 IF AN$="Y"THEN I=1:GOTO  90
70 IF AN$="N"THEN 250
80 IF AN$="L"THEN4660 ELSE 50
90 CLS:PRINT"THIS PROGRAM DRAWS ELECTRONIC    SYMBOLS AT THE POSITION OF THE
   DOT CURSOR"
100 PRINT"TO MOVE CURSOR PRESS ARROW KEYS   (SHIFT + ARROW STEPS 20)
110 PRINT:PRINT"TO DRAW LINES BETWEEN SYMBOLS               PRESS L"
120 PRINT:PRINT"TO REVERT TO CURSOR PRESS C"
130 PRINT:PRINT"TO ERASE SMALL PORTIONS PRESS E":PRINT"AND USE ARROW KEYS"
140 PRINT:PRINT"TO SEE SYMBOL GUIDE PRESS ?"
150 FOR D=1 TO 6000:NEXT
160 CLS:PRINT"SYMBOLS ARE IN 3 GROUPS
170 PRINT"SELECT SYMBOL GROUP(1,2 OR 3)"
180 PRINT:PRINT"SELECT SYMBOL(A-Y)
190 PRINT:PRINT"SELECT SCALE(1-4)
200 PRINT:PRINT "SELECT ANGLE(ARROW KEYS)
210 PRINT:PRINT"TO CLEAR LAST SYMBOL DRAWN PRESS * OR IF IN TEXT PRESS(<)
220 PRINT:PRINT"TO SELECT TEXT PRESS T
230 PRINT:PRINT"TO EXIT TEXT PRESS ENTER
240 FOR D=1 TO 10000:NEXT:IF CS$="H"THEN 310
250 PMODE4:PCLS 5:SCREEN 1,1
300 REM CURSOR CONTROL
310 PMODE4:SCREEN1,1:COLOR0,5
320 TIMER=0
330 CS$=INKEY$
340 IF CL=1 AND TIMER >1000 THEN SOUND 100,8:GOTO5970
350 S=20
360 IF CS$="0"THEN CIRCLE(X+4,Y),6,,1,.5,0
370 IF CS$="8"THEN CIRCLE(X,Y-4),6,,1,.25,.75
380 IF CS$="1"THEN H=1: GOSUB 760:GOTO 330
390 IF CS$="2"THEN H=2: GOSUB 2060:GOTO330
400 IF CS$="3"THEN H=3:GOSUB 2760:GOTO 330
410 IF CS$="C"THEN C=1:X=X+2:Y=Y+2
420 IF CS$="E"THEN C=1
430 IF CS$="L"THEN C=2
440 IF CS$=CHR$(63)AND H=1 THEN GOTO 1660
450 IF CS$=CHR$(63)AND H=2 THEN GOTO 2360
460 IF CS$=CHR$(63)AND H=3 THEN GOTO 4350
470 IF CS$=CHR$(63)AND H=0 THEN GOTO 1660
475 IF CS$="D"THEN 4660
480 IF CS$="H"THEN 90
490 IF CS$="P"THEN GOSUB 6050:GOTO 310
500 IF CS$="S"THEN GOSUB 4550:GOTO 310
510 IF CS$="T"THEN GOSUB 4760:GOTO 310
520 IF CS$=CHR$(64)THEN X=X+1:Y=Y+1:PRESET(X,Y):X=128:Y=96:PSET(X,Y):C=1
530 IF CS$=CHR$(62) THEN X=X+1:Y=Y+1:PRESET (X,Y):X=255:Y=0:PSET(X,Y):C=1
540 IF CS$=CHR$(60)THEN X=X+1:Y=Y+1:PRESET(X,Y):X=0:Y=0:PSET(X,Y):C=1
550 IF CS$=CHR$(46) THEN X=X+1:Y=Y-1: PRESET(X,Y):X=255:Y=191:PSET(X,Y):C=1
560 IF CS$=CHR$(44)THEN X=X+1:Y=Y-1:PRESET(X,Y):X=0:Y=191:PSET(X,Y):C=1
570 TU$="ABO4CD03":IF CS$=CHR$(61) THEN PLAY TU$:CLS 8:PRINT"ARE YOU SURE YOU WA
NT TO CLEAR            THE SCREEN(Y/N)":GOTO 730
580 IF CS$=CHR$(93)THEN PRESET(X,Y):X=X+20:FOR D=1 TO 500:NEXT:PSET(X,Y):IF X>25
5 THEN X=255
590 IF CS$=CHR$(21)THEN PRESET(X,Y):X=X-S:FOR D=1 TO 500:NEXT
600 IF X-S<-20 THENS=0: X=0:PSET(X,Y):GOTO330
610 IF CS$=CHR$(95)THEN PRESET(X,Y):Y=Y-S:FOR D=1 TO 500:NEXT
620 IF Y-S<-20 THEN S=0:Y=0:PSET(X,Y):GOTO 330
630 IF CS$=CHR$(91)THEN PRESET(X,Y):Y=Y+20:FOR D=1 TO 500:NEXT:PSET(X,Y):IF Y>1
1 THEN Y=190
640 XR=X:YR=Y
650 X=X+(PEEK(343)=223)-(PEEK(344)=223)+(X>255)-(X<1)
660 Y=Y+(PEEK(341)=223)-(PEEK(342)=223)+(Y>191)-(Y<1)
670 IF C<>2 THEN PRESET(XR,YR)
680 PSET(X,Y,0)
690 IF CS$=CHR$(12) AND I=1 THENCL=1:GOTO 5900
700 IF CS$=CHR$(12)THEN CL=1:GOTO 5970
```

```
710 IF CS$="*"THEN 1600
720 GOTO330
730 AN$=INKEY$:IF AN$=""THEN 730 ELSE IF AN$="N"THEN 310 ELSE IF AN$="Y" THEN PC
LS:GOTO 310
740 IF AN$<>"N"OR AN$<>"Y" THEN 730
750 REM SYMBOL DRAW STRINGS GROUP 1
760 A$="D5L5D20R5ND5R5U20L5"
770 B$="D5L5D20R5ND5R5U20L5BR6F2NF2G14"
780 C$="D5L5D10H5D5NL5D5E5D10R5ND5R5U20L5"
790 D$="D5L5BD2R5ND5R5BU2L5"
800 E$="D5R5D2L10U2R5BD4ND5NR5L5"
810 F$="D5NL5NR5BD2NR5ND5L5BL2BD6E14NH2F2"
820 G$="D5NL5R5BD2L5ND5L5BL2BD6E14NL2D2"
830 H$="D10NG10NL10NR10F10L10ND10L10"
840 I$="D10NR10F10L10ND10L10E10L10D5"
850 J$="D10NL10R10BD2L10G10R10ND10R10H10L10"
860 K$="R5U15BD5F5G5D15BU5E5NU20D10NU5H5BU10BR5R5"
870 L$="D10L10BR8BG4NL5E4R2NR10G10R10ND10R10H10"
880 M$="D5L5L5BR3G2NL3BE2R2G5NL2R10H5BR5NR12F5E5NL5L10BD5ND5R10"
890 N$="D5G5NU5D2NL5D2ND5F5NU2NL2D3"
900 O$="D5G5NU5D2NL5D2ND4NR2F5D3"
910 P$="D5F5NU5D2NR5D2ND5G5NR2NU2D5"
920 Q$="D5F5NU5D2NR5D2ND5NL2G5D5
930 R$="D5L5NU2D3NH2NG2NL5D3ND2R5D5"
940 S$="D5L5NU2D3L5NE2NF2R5D3ND2R5D5"
950 T$="D5L5NU2D4BR5ND5L5D2BL1BU3D1L5BU5BR5D1L5"
960 U$="D5NL5D4NL5D2ND4F5NU2NL2D5"
970 V$="D5NL5D4NL5D2ND4NR2F5D5"
980 W$="D5NL5NR5G5R5ND5R5H5BH8BG3ND2NR2F3BE3BH2ND2NR2F3"
990 X$="D5G5NU2D4ND2F5NU2NL2D5BU15BL10BG3NU2NL2H3BE3F3NU2NL2"
1000 Y$="D5G5NU2D4NR2ND2F5D5BU8BL10BU8NU2NL2H3BE3F3NU2L2"
1050 REM DRAW STRING SELECT ROUTINE
1060 SE$=INKEY$:IF SE$=""THEN 1060
1070 IF SE$="A"THEN Z$=A$
1080 IF SE$="B"THEN Z$=B$
1090 IF SE$="C"THEN Z$=C$
1100 IF SE$="D"THEN Z$=D$
1110 IF SE$="E"THEN Z$=E$
1120 IF SE$="F"THEN Z$=F$
1130 IF SE$="G"THEN Z$=G$
1140 IF SE$="H"THEN Z$=H$
1150 IF SE$="I"THEN Z$=I$
1160 IF SE$="J"THEN Z$=J$
1170 IF SE$="K"THEN Z$=K$
1180 IF SE$="L"THEN Z$=L$
1190 IF SE$="M"THEN Z$=M$
1200 IF SE$="N"THEN Z$=N$
1210 IF SE$="O"THEN Z$=O$
1220 IF SE$="P"THEN Z$=P$
1230 IF SE$="Q"THEN Z$=Q$
1240 IF SE$="R"THEN Z$=R$
1250 IF SE$="S"THEN Z$=S$
1260 IF SE$="T"THEN Z$=T$
1270 IF SE$="U"THEN Z$=U$
1280 IF SE$="V"THEN Z$=V$
1290 IF SE$="W"THEN Z$=W$
1300 IF SE$="X"THEN Z$=X$
1310 IF SE$="Y"THEN Z$=Y$
1350 REM SCALE SELECT ROUTINE
1360 SC$=INKEY$:IF SC$=""THEN 1360
1370 IF VAL(SC$)<1OR VAL(SC$)>4 THEN SOUND 16,8:CLS:PRINT "SCALE SELECT ERROR":P
RINT @ 166,"PRESS KEYS 1 2 3 OR 4":PRINT"REPEAT SYMBOL SELECT PROCEDURE":FOR D=1
 TO 5000:NEXT:GOTO 310
1380 IF SC$="1" THEN S=2
1390 IF SC$="2"THEN S=4
1400 IF SC$="3"THEN S=6
1410 IF SC$="4"THEN S=8
1450 REM ANGLE SELECT ROUTINE
1460 AN$=INKEY$:IF AN$=""THEN 1460
1470 A=5
1480 IF AN$=CHR$(10)THEN A=0:GOTO 1560
1490 IF AN$=CHR$(8)THEN A=1:GOTO 1560
1500 IF AN$=CHR$(94)THEN A=2:GOTO 1560
1510 IF AN$=CHR$(9)THEN A=3:GOTO 1560
1520 IF A=5 THEN SOUND16,8:CLS:PRINT"ANGLE SELECT ERROR":PRINT @ 162,"PRESS ONE
OF THE ARROW KEYS":PRINT"REPEAT SYMBOL SELECT PROCEDURE":FOR D=1 TO 4000:NEXT:GO
TO 310
1550 REM SYMBOL FINAL DRAW COMMAND ROUTINE
1560 FD$="S"+STR$(S)+"A"+STR$(A)+Z$
1570 DRAW"C0;BM"+STR$(X)+","+STR$(Y)+";XFD$;"
1580 FOR D=1 TO 500:NEXT
1590 RETURN
1600 DRAW"C5;BM"+STR$(X)+","+STR$(Y)+";XFD$;":DRAW"C0":GOTO330
1650 REM SYMBOL LETTER GUIDE
1660 CLS:PRINT @ 2,"SYMBOL LETTER GUIDE GROUP 1."
1670 PRINT @64,"A=RESISTOR"
1680 PRINT @ 96,"B=PRESET"
```

```
1690 PRINT @ 128,"C=POTENTIOMETER"
1700 PRINT @ 160,"D=CAPACITOR"
1710 PRINT @ 192,"E=ELECTROLYTIC CAPACITOR"
1720 PRINT @ 224,"F=TRIMMER"
1730 PRINT @ 256,"G=VARIABLE CAPACITOR"
1740 PRINT @ 288,"H=DIODE"
1750 PRINT @ 320,"I=ZENER DIODE"
1760 PRINT @ 352,"J=VARICAP DIODE"
1770 PRINT @ 384,"K=DIAC"
1800 REM NEXT PAGE OR RETURN TO GRAPHICS
1810 PRINT:PRINT"TO SEE NEXT PAGE PRESS N"
1820 FOR D=1 TO 500
1830 N$=INKEY$:IF N$<>"N"AND D<>500  THEN NEXT:IF N$<>"N"AND D=500 THEN GOTO 185
0
1840 IF N$="N"THEN CLS:GOTO1860
1850 GOTO 310
1860 PRINT @ 0,"L=THYRISTOR"
1870 PRINT @ 32,"M=TRIAC"
1880 PRINT @ 64,"N=NPN TRANSISTOR(FACING RIGHT)"
1890 PRINT @ 96,"O=PNP TRANSISTOR(FACING RIGHT)
1900 PRINT @ 128,"P=NPN TRANSISTOR(FACING LEFT)
1910 PRINT @ 160,"Q=PNP TRANSISTOR(FACING LEFT)
1920 PRINT @ 192,"R=FET(N CHANNEL)
1930 PRINT @ 224,"S=FET(P CHANNEL)
1940 PRINT @ 256,"T=DUAL GATE FET
1950 PRINT @ 288,"U=UNIJUNCTION(N BASE)
1960 PRINT @ 320,"V=UNIJUNCTION(P BASE)
1970 PRINT @ 352,"W=LIGHT EMITTING DIODE"
1980 PRINT @ 384,"X=NPN PHOTO TRANSISTOR"
1990 PRINT @ 416,"Y=PNP PHOTO TRANSISTOR"
2000 FOR D=1 TO 4000:NEXT:IF H=0 THEN 2360 ELSE 310
2050 REM SYMBOL DRAW STRINGS GROUP 2
2060 A$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3L10U3NL3U3NL3U3NL3U3BR2R2L1U2G1"
2070 B$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3L10U3NL3U3NL3U3NL3U3NL3U3N
3U3NL3U3BR2R2L1U2G1"
2080 C$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3L10U3NL3U3NL3U3NL3U3N
3U3NL3U3NL3U3NL3U3BR2R2L1U2G1"
2090 D$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3L10U3NL3U3NL3U3N
3U3NL3U3NL3U3NL3U3NL3U3NL3U3BR2R2L1U2G1"
2100 E$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3L10U3NL3U3N
3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3BR2R2L1U2G1"
2110 F$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3L10U3N
3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3BR2R2L1U2G1"
2120 G$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3L
0U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3BR2R2L1U2G1"
2130 H$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3N
3D3NR3D3L10U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3N
3U3BR2R2L1U2G1"
2140 I$="R3U3R10D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3N
3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3NR3D3L10U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3N
3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3NL3U3BR2R2L1U2G1"
2150 J$="G5NU2D4NF5D2L5U8R5"
2160 K$="D5L5D3R5ND5R5U3L5"
2170 L$="F2G2NU4D4BR6U8G2F2"
2180 M$="F2G2NU4ND6BR6H2E2ND10F2G2BR4H2E2D10"
2190 N$="D5L3D6R3ND5R3U6NL3G6U6F6"
2200 O$="R5BU2L2R3U3NL1NR1D3R3BD2L2R5
2210 P$="R2E2BD2R5"
2220 Q$="H3ND1NR1BL2U1L1D1R1BR10U1L1D1R1"
2230 R$="H3ND1NR1BL1U1L1D1R1BU4BR4U1L1D1R1BD4BR4U1L1D1R1
2240 S$="H3ND1NR1BL2U1L1D1R1BU4BR4U1L1D1R1BR4U1L1D1R1BD4BR4U1L1D1R1
2250 T$="H3ND1NR1BL2U1L1D1R1BU2BR2U1L1D1R1BU2BR2U1L1D1R1BR4U1L1D1R1BD2BR2U1L1D1
1BD2BR2U1L1D1R1"
2260 U$="D3NH1NE1BD3BL1U1L1D1R1BL2BU1U1L1D1R1BU2BL2U1L1D1R1BU2U1L1D1R1BU2BR1U1L
D1R1BU2BR2U1L1D1R1BU1BR2U1L1D1R1BR2U1L1D1R1BD1BR2U1L1D1R1BD2BR1U1L1D1R1BR1BD2U1
1D1R1BD2U1L1D1R1"
2270 V$="R5U5R10D5NR5D5L5BH4ND1NR1F4L5U5"
2280 W$="D5NL3R3BR2BD1L10D2R10NU2BD1BL2L3ND5L3"
2290 X$="BU12F5NE2NG2D4E4L4BF2F5G5NH2NF2L4F4U4BG2G5H5E2L4D4E2BH2NE2NG2H5E5F2U4L
F2BE2NH2NF2E5"
2300 Y$="D3NH3NE3D5"
2310 GOSUB 1060
2320 GOTO 330
2350 REM SYMBOL LETTER GUIDE 2
2360 CLS:PRINT @ 2,"SYMBOL LETTER GUIDE GROUP 2."
2370 PRINT @ 64,"A=EIGHT PIN I.C."
2380 PRINT @ 96,"B=FOURTEEN PIN I.C."
2390 PRINT @ 128,"C=SIXTEEN PIN I.C."
2400 PRINT @ 160,"D=EIGHTEEN PIN I.C."
2410 PRINT @ 192,"E=TWENTY PIN I.C."
2420 PRINT @ 224,"F=TWENTYTWO PIN I.C."
2430 PRINT @ 256,"G=TWENTYFOUR PIN I.C."
2440 PRINT @ 288,"H=TWENTYEIGHT PIN I.C."
2450 PRINT @ 320,"I=FORTY PIN I.C."
2460 PRINT @ 352,"J=LOUDSPEAKER"
2470 PRINT @ 384,"K=RELAY COIL"
2500 REM NEXT PAGE OR RETURN TO GRAPHICS
2510 PRINT:PRINT"TO SEE NEXT PAGE PRESS N"
2520 FOR D=1 TO 500
2530 N$=INKEY$:IF N$<>"N"AND D<>500 THEN NEXT:
IF N$<>"N"AND D=500 THEN GOTO 2550
2540 IF N$="N" THEN CLS:GOTO 2560
2550 GOTO 310
2560 PRINT @ 0,"L=SINGLE POLE RELAY CONTACTS"
2570 PRINT @ 32,"M=DOUBLE POLE RELAY CONTACTS"
2580 PRINT @ 64,"N=TRANSMISSION GATE"
2590 PRINT @ 96,"O=PUSH TO MAKE SWITCH"
2600 PRINT @ 128,"P=SINGLE POLE SWITCH"
2610 PRINT @ 160,"Q=S.P.D.T.SWITCH"
2620 PRINT @ 192,"R=ONE POLE THREE WAY SWITCH"
2630 PRINT @ 224,"S=ONE POLE FOUR WAY SWITCH"
2640 PRINT @ 256,"T=ONE POLE SIX WAY SWITCH"
2650 PRINT @ 288,"U=ONE POLE TWELVE WAY SWITCH"
2660 PRINT @ 320,"V=METER"
```

```
2670 PRINT @ 352,"W=CRYSTAL"
2680 PRINT @ 384,"X=BRIDGE RECTIFIER"
2690 PRINT @ 416,"Y=ANTENNA"
2700 FOR D=1 TO 5000:NEXT:IF H=0 THEN 4350 ELSE 310
2750 REM WOUND COMPONENTS SELECTION ROUTINE
2760 W$=INKEY$:IF W$=""THEN 2760
2770 W=ASC(W$):IF W<65 OR W>76THEN CLS:SOUND 16,8:
PRINT"SELECTION ERROR":PRINT@ 162,"PRESS A TO L":
PRINT"REPEAT SELECTION PROCEDURE":FOR D=1 TO 4000:
NEXT:GOTO 310
2780 IF W=65 THEN GOSUB 2900:GOTO330
2790 IF W=66 THEN GOSUB 3000:GOTO 330
2800 IF W=67 THEN GOSUB 3100:GOTO330
2810 IF W=68 THEN GOSUB 3250:GOTO330
2820 IF W=69 THEN GOSUB 3350:GOTO330
2830 IF W=70 THEN GOSUB 3450:GOTO330
2840 IF W=71 THEN GOSUB 3550:GOTO330
2850 IF W=72 THEN GOSUB 3700:GOTO330
2860 IF W=73 THEN GOSUB 3850:GOTO330
2870 IF W=74 THEN GOSUB 3950:GOTO330
2880 IF W=75 THEN GOSUB 4050:GOTO330
2890 IF W=76 THEN GOSUB 4200:GOTO330
2900 REM COIL SUBROUTINE
2910 I=X-2:J=Y+7:PRESET(X,Y)
2920 FOR L=1 TO 3
2930 CIRCLE(X,Y),3,,1,.75,.25
2940 CIRCLE(I,J),3,,1,.25,.75
2950 J=J+12:Y=Y+12
2960 NEXT L
2970 RETURN
3000 REM COIL SUBROUTINE(HORIZONTAL)
3010 I=X+7:J=Y+1:PRESET(X,Y)
3020 FOR L=1 TO 3
3030 CIRCLE(X,Y),3,,1,.5,0
3040 CIRCLE(I,J),3,,1,0,.5
3050 I=I+12:X=X+12
3060 NEXT L
3070 RETURN
3100 REM AIR CORED TRANSFORMER
3110 I=X-1:J=Y+7:PRESET(X,Y)
3120 FOR L=1 TO 3
3130 CIRCLE(X,Y),3,,1,.75,.25
3140 CIRCLE(X+10,Y),3,,1,.75,.25
3150 CIRCLE(I,J),3,,1,.25,.75
3160 CIRCLE(I+10,J),3,,1,.25,.75
3170 Y=Y+12
3180 J=J+12
3190 NEXT L
3200 RETURN
3250 REM AIR CORE TRANSFORMER(HORIZONTAL)
3260 I=X+7:J=Y+1:PRESET(X,Y)
3270 FOR L=1 TO 3
3280 CIRCLE(X,Y),3,,1,.5,0
3290 CIRCLE(I,J),3,,1,0,.5
3300 CIRCLE(X,Y-10),3,,1,.5,0
3310 CIRCLE(I,J-10),3,,1,0,.5
3320 I=I+12:X=X+12
3330 NEXT L
3340 RETURN
3350 REM FERRITE CORED COIL
3360 I=X-1:J=Y+7:PRESET(X,Y)
3370 FOR L=1 TO 3
3380 CIRCLE(X,Y),3,,1,.75,.25
3390 CIRCLE(I,J),3,,1,.25,.75
3400 J=J+12:Y=Y+12
3410 NEXT L
3420 DRAW"S4;A0;BM"+STR$(X-1)+","+STR$(Y)+
";BU6U28BL8BD22E16NL3D3;"
3430 RETURN
3450 REM FERRITE CORED COIL(HORIZONTAL)
3460 I=X+7:J=Y+1:PRESET(X,Y)
3470 FOR L=1 TO 3
3480 CIRCLE(X,Y),3,,1,.5,0
3490 CIRCLE(I,J),3,,1,0,.5
3500 I=I+12:X=X+12
3510 NEXT L
3520 DRAW"S4;A0;BM"+STR$(X)+","+STR$(Y)+
";BL6L28BD8BR4E16NL3D3;"
3530 RETURN
3550 REM I.F.TRANSFORMER
3560 I=X-1:J=Y+7:PRESET(X,Y)
3570 FOR L=1 TO 3
3580 CIRCLE(X,Y),3,,1,.75,.25
3590 CIRCLE(X+12,Y),3,,1,.75,.25
3600 CIRCLE(I,J),3,,1,.25,.75
3610 CIRCLE(I+12,J),3,,1,.25,.75
3620 Y=Y+12:J=J+12
```

```
3630 NEXT L
3640 LINE(X+5,Y-2)-(X+5,Y-38),PSET
3650 DRAW"S4;A0;BM"+STR$(X-7)+","+STR$(Y-7)+
";E24NL3D3;"
3660 RETURN
3700 REM I.F.TRANSFORMER(HORIZONTAL)
3710 I=X+7:J=Y+1:PRESET(X,Y)
3720 FOR L=1 TO 3
3730 CIRCLE(X,Y),3,,1,.5,0
3740 CIRCLE(X,Y-12),3,,1,.5,0
3750 CIRCLE(I,J),3,,1,0,.5
3760 CIRCLE(I,J-12),3,,1,0,.5
3770 I=I+12:X=X+12
3780 NEXT L
3790 LINE(X,Y-5)-(X-38,Y-5),PSET
3800 DRAW"S4;A0;BM"+STR$(X-36)+","+STR$(Y+5)+
";E26NL3D3;"
3810 RETURN
3850 REM IRON CORRED CHOKE
3860 I=X-1:J=Y+7:PRESET(X,Y)
3870 FOR L=1 TO 3
3880 CIRCLE(X,Y),3,,1,.75,.25
3890 CIRCLE(I,J),3,,1,.25,.75
3900 J=J+12:Y=Y+12
3910 NEXT L
3920 LINE(X-1,Y-6)-(X-1,Y-34),PSET
3930 RETURN
3950 REM IRON CORED CHOKE(HORIZONTAL)
3960 I=X+7:J=Y+1:PRESET(X,Y)
3970 FOR L=1 TO 3
3980 CIRCLE(X,Y),3,,1,.5,0
3990 CIRCLE(I,J),3,,1,0,.5
4000 I=I+12:X=X+12
4010 NEXT L
4020 LINE(X-34,Y+1)-(X-6,Y+1),PSET
4030 RETURN
4050 REM LOW FREQUENCY TRANSFORMER
4060 I=X-1:J=Y+7:PRESET(X,Y)
4070 FOR L=1 TO 5
4080 CIRCLE(X,Y),3,,1,.75,.25
4090 CIRCLE(X+12,Y),3,,1,.75,.25
4100 CIRCLE(I,J),3,,1,.25,.75
4110 CIRCLE(I+12,J),3,,1,.25,.75
4120 Y=Y+12:J=J+12
4130 NEXT L
4140 LINE(X+5,Y-2)-(X+5,Y-58),PSET
4150 RETURN
4200 REM LOW FREQUENCY TRANSFORMER(HORIZONTAL)
4210 I=X+7:J=Y+1:PRESET(X,Y)
4220 FOR L=1 TO 5
4230 CIRCLE(X,Y),3,,1,.5,0
4240 CIRCLE(I,J),3,,1,0,.5
4250 CIRCLE(X,Y-12),3,,1,.5,0
4260 CIRCLE(I,J-12),3,,1,0,.5
4270 I=I+12:X=X+12
4280 NEXT L
4290 LINE(X-58,Y-5)-(X-6,Y-5),PSET
4300 RETURN
4350 REM WOUND COMPONENTS GUIDE
4360 CLS:PRINT TAB(8)"WOUND COMPONENTS"
4370 PRINT@32,"A=COIL (V)"
4380 PRINT@64,"B=COIL (H)"
4390 PRINT@96,"C=TRANSFORMER(AIR CORE) (V)
4400 PRINT@128,"D=TRANSFORMER(AIR CORE) (H)"
4410 PRINT@160,"E=COIL(FERRITE CORE) (V)"
4420 PRINT@192,"F=COIL(FERRITE CORE) (H)"
4430 PRINT@224,"G=I.F.TRANSFORMER (V)"
4440 PRINT@256,"H=I.F.TRANSFORMER (H)
4450 PRINT@288,"I=CHOKE (V)"
4460 PRINT@320,"J=CHOKE (H)"
4470 PRINT@352,"K=L.F.TRANSFORMER (V)"
4480 PRINT@384,"L=L.F.TRANSFORMER (H)"
4490 PRINT"A BRIDGE OVER WIRES IS DRAWN
WITH 8 FOR (V) OR 0 (H)
4500 FOR D=1 TO 6000:NEXT:GOTO 310
4550 REM CASSETTE SAVE ROUTINE
4560 CLS2:PRINT @ 32,"    SAVE DIAGRAM TO CASSETTE"
4570 AUDIO ON:MOTOR ON
4580 PRINT:PRINT" POSITION TAPE READY TO RECORD
(motor and audio are on)": PRINT:PRINT"
PRESS enter WHEN READY"
4590 R$=INKEY$:IF R$<>CHR$(13)THEN4590
ELSE MOTOR OFF
4600 PRINT:INPUT"    FILENAME(8 CHARACTERS MAX)
";NA$:MOTOR OFF:PRINT:PRINT"PLACE RECORDER IN
RECORD MODE":FOR D= 1 TO 10000:NEXT:CLS 3:PRINT
@ 32,"    TO START SAVING ";NA$:PRINT @ 64,"
PRESS S"
```

```
4610 S$=INKEY$:IF S$<>"S"THEN 4610
4620 IF S$="S"THEN CLS 4:PRINT @ 32,"         SAVING ";NA$ :FOR D=1 TO2000:NEXT
4630 CSAVEM NA$,1536,7679,1536
4640 RETURN
4650 REM CASSETTE LOADING ROUTINE
4660 CLS 3
4670 MOTOR ON:AUDIO ON:PRINT @ 160," POSITION TAPE READY TO LOAD       (motor an
d audio are on)          THEN PRESS enter"
4680 T$=INKEY$:IF T$<>CHR$(13)THEN 4680 ELSE MOTOR OFF:AUDIO OFF
4690 FOR D=1 TO 1000:NEXT:CLS 3:INPUT"FILENAME OF DIAGRAM REQUIRED    ";NA$:CLS
3:PRINT @ 160,"          LOADING " ;NA$:FOR D=1 TO 1000:NEXT:PMODE 4:PCLS 5:SC
REEN 1,1:CLOADM NA$
4700 GOTO 310
4750 REM TEXT ROUTINE
4760 CLS 7:PRINT @ 32,"          TEXT ROUTINE":PRINT @ 96,"         SELECT VERTIC
AL (V)          OR HORIZONTAL (H) PRINT":N=-8
4770 CT$=INKEY$:IF CT$=""THEN 4770
4780 IF CT$="H"OR CT$="V"THEN 4790 ELSE PRINT"INVALID INPUT PRESS (H) OR (V)":GO
TO 4770
4790 SCREEN 1,1:PRESET(X,Y)
4800 ZZ$=INKEY$:IF ZZ$=""THEN 4800
4810 NU$="U3NG1D6L1R2L1U3":IF ZZ$="1"THEN ZZ$=NU$
4820 NU$="G2D1R4L2BU3R1E1U1H1L2G1BF2":IF ZZ$="2"THEN ZZ$=NU$
4830 NU$="R1F1D1G1L2H1BU4E1R2F1D1G1L1":IF ZZ$="3"THEN ZZ$=NU$
4840 NU$="BF1D2U6G3D1R4L2BU1":IF ZZ$="4"THEN ZZ$=NU$
4850 NU$="BG2F1R2E1U2H1L3U2R4L2BD3":IF ZZ$="5"THEN ZZ$=NU$
4860 NU$="L2R3F1D1G1L2H1U3E2R1L1BD3":IF ZZ$="6"THEN ZZ$=NU$
4870 NU$="G1ND2E3U1L4R2BD3":IF ZZ$="7"THEN ZZ$=NU$
4880 NU$="L1G1D1F1R2E1U1H1L2H1U1E1R2F1D1G1L1":IF ZZ$="8"THEN ZZ$=NU$
4890 NU$="R2L3H1U1E1R2F1D3G2L1R1BU3":IF ZZ$="9"THEN ZZ$=NU$
4900 NU$="G2F1R2E1U4NG2H1L2G1D4E2":IF ZZ$="0"THEN ZZ$=NU$
4910 A$="L2D3U5E1R2F1D2ND3L2":IF ZZ$="A"THEN ZZ$=A$
4920 B$="L1U3L1R3F1D1G1L2D3L1R3E1U1H1L1":IF ZZ$="B"THEN ZZ$=B$
4930 C$="BE2H1L2G1D4F1R2E1BH2":IF ZZ$="C"THEN ZZ$=C$
4940 D$="BL2U3R2F2D2G2L2U3BR2":IF ZZ$="D"THEN ZZ$=D$
4950 E$="R1L3D3R4BU6L4D3R2":IF ZZ$="E"THEN ZZ$=E$
4960 F$="R1L3D3U6R4L2BD3":IF ZZ$="F"THEN ZZ$=F$
4970 G$="BE2H1L2G1D4F1R3U2L1BH1":IF ZZ$="G"THEN ZZ$=G$
4980 H$="L2D3BR4U6BL4D3R4L2":IF ZZ$="H"THEN ZZ$=H$
4990 I$="D3L1R2BU6L2R1D3":IF ZZ$="I"THEN ZZ$=I$
5000 J$="BG2F1R1E1U5R1L2BD3":IF ZZ$="J"THEN ZZ$=J$
5010 K$="BL1E3BD6H3L1U3D6U3BR2":IF ZZ$="K"THEN ZZ$=K$
5020 L$="BL2U3D6R4L2BU3":IF ZZ$="L"THEN ZZ$=L$
5030 M$="U1H2D6BR4U6G2D1":IF ZZ$="M"THEN ZZ$=M$
5040 N$="H2UD6BR4U6D5H2":IF ZZ$="N"THEN ZZ$=N$
5050 O$="BH2E1R2F1D4G1L2H1U4BF2":IF ZZ$="O"THEN ZZ$=O$
5060 P$="L2D3U6R3F1D1G1L1":IF ZZ$="P"THEN ZZ$=P$
5070 Q$="BD1F2BL2E2U3H1L2G1D4F1R1BU3":IF ZZ$="Q"THEN ZZ$=Q$
5080 R$="L1F3BL4U3R3E1U1H1L3D3R2":IF ZZ$="R"THEN ZZ$=R$
5090 S$="L1H1U1E1R2F1BG2R1F1D1G1L2H1BE2":IF ZZ$="S"THEN ZZ$=S$
5100 T$="D3U6L2R4L2D3":IF ZZ$="T"THEN ZZ$=T$
5110 U$="BL2U3BR4D5G1L2H1U2BR2":IF ZZ$="U"THEN ZZ$=U$
5120 V$="BD2ND1E1U1E1U2BL4D2F1D1F1BU2":IF ZZ$="V"THEN ZZ$=V$
5130 W$="D1F2U6BL4D6E2U1":IF ZZ$="W"THEN ZZ$=W$
5140 X$="E2U1BL4D1F4D1BL4U1E2":IF ZZ$="X"THEN ZZ$=X$
5150 Y$="E2U1BL4D1F2ND3":IF ZZ$="Y"THEN ZZ$=Y$
5160 Z$="E2U1L4R4BD6L4U1E2":IFZZ $="Z"THEN ZZ$=Z$
5200 REM SPECIAL TEXT SYMBOL STRINGS
5210 SP$="BU3G2D2F2BU3":IF ZZ$="("THEN ZZ$=SP$
5220 SP$="BU3F2D2G2BU3":IF ZZ$=")"THEN ZZ$=SP$
5230 SP$="L2BF2U4BF2L2":IF ZZ$="+"THEN ZZ$=SP$
5240 SP$="L2R4L2":IF ZZ$="-"THEN ZZ$=SP$
5250 SP$="L2R4L2BU2U1BD5D1BU2":IF ZZ$="/"THEN ZZ$=SP$
5260 SP$="NG2NF2NE2NH2":IF ZZ$="*"THEN ZZ$=SP$
5270 SP$="BR8":IF ZZ$=CHR$(32)THEN ZZ$=SP$
5280 SP$="R1":IF ZZ$=CHR$(46)THEN ZZ$=SP$
5290 SP$="BL1BD3U4BR4D4L4G2":IF ZZ$="!"THEN ZZ$=SP$
5300 SP$="BL1D3BR3U3H1L1G1BR1":IF ZZ$=CHR$(34)THEN ZZ$=SP$
5310 SP$="BL3ND3E1R1F1ND2E1R1F1D3":IFZZ$="£"THEN ZZ$=SP$
5320 SP$="BD2BL6R2U2E2R2F2D2R2":IF ZZ$="$"THEN ZZ$=SP$
5330 SP$="D2G2D2F2D2":IF ZZ$="%"THEN ZZ$=SP$
5340 SP$="U1E2R2F2D2F2R2E2U2":IF ZZ$="&"THEN ZZ$=SP$
5350 SP$="BR6BD2R2U4R3D4R2":IF ZZ$="'"THEN ZZ$=SP$
5360 SP$="E8L3F3U3":IF ZZ$=CHR$(95)THEN ZZ$=SP$
5370 SP$="H8R3G3U3":IF ZZ$=CHR$(21)THEN ZZ$=SP$
5380 SP$="U11G3R6H3":IF ZZ$=CHR$(94)THEN ZZ$=SP$
5390 SP$="D11H3R6G3":IF ZZ$=CHR$(10)THEN ZZ$=SP$
5400 SP$="L11E3D6H3":IF ZZ$=CHR$(8)THEN ZZ$=SP$
5410 SP$="R11H3D6E3":IF ZZ$=CHR$(9)THEN ZZ$=SP$
5420 SP$="G8U3F3L3":IF ZZ$=CHR$(91)THEN ZZ$=SP$
5430 SP$="F8U3G3R3":IF ZZ$=CHR$(93)THEN ZZ$=SP$
5440 SP$="BD2BR2E3F3E3F3":IF ZZ$="@"THEN ZZ$=SP$
5450 SP$="BU1L2R4BD2L4R2BU1":IF ZZ$="="THEN ZZ$=SP$
5460 IF ZZ$=":"THEN SOUND 18,8:SOUND 1,8:GOTO 4800
5470 IF ZZ$=CHR$(12)THEN SOUND 18,8:SOUND 1,8:GOTO 4800
5480 IF ZZ$=","THEN SOUND 18,8:SOUND 1,8:GOTO 4800
5490 IF ZZ$=";"AND CT$="H"THEN PSET(X+N+4,Y):FOR D=1 TO 500:NEXT:PRESET(X+N+4,Y)
:GOTO4800
```

```
5500 IF ZZ$=";"AND CT$="V"THEN PSET(X,Y+N+8):FOR D=1 TO 500:NEXT:PRESET(X,Y+N+8)
:GOTO 4800
5510 IF ZZ$="?"THEN GOSUB 5710:ZZ$=CHR$(13)
5520 IF ZZ$=CHR$(13)THEN N=-8:RETURN
5530 SP$="BL4BD4U8R8D1L8D1R8D1L8D1R8D1L8D1R8D1L8D1R8":IF CT$="V" AND ZZ$="<"THEM
 ZZ$=SP$:GOTO5650
5540 SP$="BL4BU4D8R1U8R1D8R1U8R1D8R1U8R1D8R1U8":IF CT$="H" AND ZZ$="<"THEN ZZ$=S
P$:GOTO 5630
5550 SP$="BL5BU5R12D10L12U10BD2BR2R2ND6R2BR2ND6R2F1D1G1L2":IF ZZ$=">"THEN ZZ$=SP
$
5560 N=N+8:IF CT$="V"THEN 5600
5570 IF X+N>250 THEN X=2:N=0:Y=Y+8:SOUND 200,16
5580 DRAW"C0;S4;A0;BM"+STR$(X+N)+","+STR$(Y)+";XZZ$;"
5590 IF CT$="H" THEN GOTO 4800
5600 IF Y+N>185 THEN SOUND 200,16: RETURN
5610 DRAW "C0;S4;A0;BM"+STR$(X)+","+STR$(Y+N)+";XZZ$;"
5620 GOTO4800
5630 IF SGN (X+N)=-1 THEN X=250:Y=Y-8:N=0
5640 DRAW"C5;S4;A0;BM"+STR$(X+N)+","+STR$(Y)+";XZZ$;":X=X-8:GOTO 4800
5650 DRAW"C5;S4;A0;BM"+STR$(X)+","+STR$(Y+N)+";XZZ$;":Y=Y-8:GOTO4800
5660 FOR D=1 TO 5000:NEXT:RETURN
5700 REM TEXT INFORMATION
5710 CLS:PRINT @ 160,"THE NUMBER KEYS USED WITH SHIFT KEY DRAW THE FOLLOWING SYM
BOLS"
5720 FOR D =1 TO 5000:NEXT:CLS
5730 PRINT:PRINT"MICRO:NANO:MILLI:OHMS HORIZONTAL:OHMS VERTICAL:SINE WAVEFORM:
 SQUARE WAVEFORM:OPEN BRACKET:    CLOSED BRACKET."
5740 PRINT:PRINT"THE ARITHMETIC SYMBOLS ARE DRAWN WITH THE (+),(-),(*)AND(/)KEYS
."
5750 PRINT"NORTH,SOUTH,EAST AND WEST ARROWS ARE DRAWN WITH ARROW KEYS."
5760 PRINT:PRINT"NORTH EAST,SOUTH EAST,SOUTH WEST AND NORTH WEST ARROWS ARE DRAW
N WITH THE SHIFT + ARROW KEYS.     TO SEE NEXT PAGE PRESS (N)."
5770 FOR D=1 TO 1000
5780 N$=INKEY$:IF N$<>"N"AND D<>1000 THEN NEXT:IF N$<>"N"AND D=1000 THEN GOTO 58
00
5790 IF N$="N"THEN CLS:GOTO 5810
5800 RETURN
5810 PRINT"THE @ KEY DRAWS A SAWTOOTH              WAVEFORM."
5820 PRINT:PRINT"THE(,)KEY DRAWS A RAMP WAVEFORM."
5830 PRINT"THE(>)KEY DRAWS A TEST POINT BOX"
5840 PRINT:PRINT"THE (;) KEY SHOWS THE CURSOR       POSITION FOR A BRIEF MOMENT"
5850 FOR D=1 TO 10000:NEXT:RETURN
5900 REM CLEAR LARGE SECTION OF DIAGRAM
5910 CLS:PRINT" LARGE SCALE WIPEOUT ROUTINE ":PRINT:PRINT"MOVE CURSOR TO TOP RIG
HT OF ":PRINT"PORTION TO BE WIPED OUT,PRESS ":PRINT"THE DOWN ARROW KEY UNTIL LIN
E":PRINT"REACHES BOTTOM OF PORTION TO ":PRINT"BE WIPED OUT,THEN PRESS LEFT "
5920 PRINT TAB(11)"ARROW KEY"
5930 PRINT:PRINT"  A TONE WILL SOUND WHEN THE       CURSOR MOVE TIME HAS ELAPSED,
  THEN PRESS THE DOWN ARROW KEY"
5940 FOR D=1 TO 8000:NEXT:SCREEN1,1
5950 N=0
5960 TIMER=0:GOTO 330
5970 IF PEEK(342)=223 THEN N=N+1:Z=Y-N:GOTO 5980 ELSE 5990
5980 Y=Y+1:PSET(X,Y):IF Y=191 THEN 330
5990 IF PEEK(343)=223 THEN 6000 ELSE 5970
6000 X=X-1:IF SGN(Z)=-1THEN Z=0
6010 LINE(X+1,Z)-(X+1,Y),PRESET
6020 IF PEEK(343)=255 THEN CL=0:GOTO330
6030 IF SGN(X)=-1 THEN 330 ELSE GOTO6000
6050 REM PRINT DIAGRAM ROUTINE
6060 CLS:PRINT"REMOVE DATA TAPE AND REPLACE IT WITH cirdraw POSITIONED AT STARTO
F SCREEN DUMP PROGRAM":PRINT:PRINT"  TO OBTAIN HARD COPY THE NEXT  PROGRAM WILL
BE LOADED FROM THE              TAPE"
6070 PRINT:PRINT" MAKE SURE YOU HAVE ALL DETAILS CORRECT AS THIS PROGRAM WILL BE
          DELETED"
6080 PRINT:PRINT" TO CONTINUE PRESS (C) OR TO      RETURN TO DRAW ROUTINE PRESS (R
)"
6090 C$=INKEY$:IF C$=""THEN 6090
6100 IF C$="R"THEN RETURN
6110 IF C$<>"C"THEN 6090
6120 IF C$="C"THEN CLS4:PRINT@128,CHR$(191)+CHR$(191)"LOADING SCREEN DUMP PROGRA
M"CHR$(191)+CHR$(191)+CHR$(191);:PRINT STRING$(32,191);:PRINT STRING$(11,191)"PL
EASE WAIT"STRING$(11,191);
6130 CLOADM"CIRDRAW2":NEW
```

Write: ADVENTURE

Pete Gerrard forgets about programming and starts looking for stories

RECENTLY I have been going through my collection of Isaac Asimov science fiction novels (again!), and the last two that I reread were *The Gods Themselves* and *The Currents of Space*. Perhaps a better description of these two particular books would be science faction, as both take a plausible, or at least extremely well hypothesised, piece of scientific knowledge and weave around it an exciting tale of interstellar and intergalactic skullduggery, with just a little bit of romance thrown in for good measure.

It has often struck me that the science fiction world of adventure games is sadly lacking, although Infocom come to our rescue (as usual on mightier machines than the Dragon) with such gems as Planetfall and Stationfall, both featuring the truly wonderful robotic creation known as Floyd. Some positronic brains do 'ave 'em, although I can't see Floyd ever taking the lead role in The Phantom of the Opera somehow.

The point linking those two paragraphs is this: why don't more adventure writers take their example from dear old Isaac's methods, and concoct an exciting adventure based in some way on scientific fact, or something that has repeatedly proposed as scientific fact. Remember, no-one has for certain found a black hole yet, no-one has for certain proved that quarks and gluons exist, but everybody tends to accept them as being essential to proving various theories.

Thus we arrive at a paraphrasing of Infocom's usual term: Interactive Faction, adventures that start off with a sound basing in fact, or proposed fact, and which take those facts as the basis for an interesting, possibly humorous, certainly different, type of adventure game.

With all the lofty eminence of a degree in astronomy from University College London (albeit from years ago ... ageing hippy, Helen, I know!), *(Join the UCL old bats club.)* I think it's time to wander through and around one or two science fiction 'stories' that could easily be used as the basis for an adventure game, starting with something very close to the plot behind the aforementioned Asimov classic **The Gods Themselves**.

Bar, library

In my first year at university we all had to write a report on any 'unusual' aspect of astronomy that interested us, and we all went scurrying to the library (well, to be honest, bar first, library later) in search of information and inspiration. It was there that I first read all about black holes, worm holes, and white holes. What? Virtually everyone has heard of black holes, but the others? They have been proposed in several different scientific journals, by a variety of different authors. The theory is simple. If we take the postulate that a black hole exists, then what happens to all that energy that's being sucked into the things? It can't just vanish, that would be breaking several laws of physics, and apart from anything else it would leave us with a universe that was gradually running out of energy. Entropy, and all that.

So, the theory is that it all travels along a worm hole until it re-emerges via a white hole into another universe. However, if this is the case, then other universes must also have black holes which are sending energy to us via worm holes and white holes, in order to maintain stability. Why has nobody detected any white holes? Nobody has detected any black holes yet, although there is a strong case for one in the constellation of Cygnus, the Swan. A beautifully constructed theory that really explains nothing, but which opens up the path to other universes.



Parallel universes, alternate universes, call them what you will, but an adventure that starts off with our explorer vanishing down a black hole and emerging into another universe, then desperately trying to get back to his own place and time, would be an interesting one. In an adventure like that, just think of the fun you could have dreaming up your other universe, where none of the laws of physics as we know them would necessarily apply. What might be an exceedingly heavy object in our universe might be very light in another one, although it would still possess the same inertia. Well, possibly, anyway, you might want to change a little bit more than just the laws of gravity.

I briefly looked at some of Carl Sagen's more outrageous writings on the subject of life on other planets, and in particular the planet Jupiter. No-one will be able to prove his theories right or wrong for many years, perhaps centuries, because Jupiter is a difficult place to explore to say the least. However, an adventurer could go there, and meet all the strange, weird and wonderful animals that Sagen puts forward as possibly existing in the thick Jovian atmosphere.

The tenth planet

What I eventually concentrated on was the search for the tenth planet. Arthur C. Clarke, another superb science faction author, repeatedly makes mention of a tenth planet called Persephone, but what he intends it to be is uncertain, as it appears to have no purpose other than dating a story of his as some way off in the future after this tenth planet has been found. According to all the scientific papers I read, there is a stronger case for arguing that the Sun is part of a binary system, with its partner being way, way beyond the orbits of Pluto and Neptune. This 'star' is supposed to be visible in the infra-red, is supposed to have run out of steam long ago, and could explain some slight perturbations still to be found in the orbits of the outer planets.

Send an adventurer there, that's what I say, and let him explore this companion star of our Sun. Perhaps it's not a part of nature at all, but some ancient artefact put there by explorers long ago. Not very scientifically plausible, but in the world of adventures anything goes. Nobody, incidentally, has yet proved that there isn't another star out there in some sort of linked orbit with our Sun: there's a stronger case for its existence than there is for its non-existence. This is the sort of situation where, once again, you could let your imagination run riot and create a whole new world for your adventurers to explore. If any of you have ever read Arthur C. Clarke's *Rendezevous with Rama* you'll know the sort of thing. Not only are you exploring an unknown world, but you're also exploring one created by an unknown intelligence, one that has a completely different way of thinking to mankind. Thus you could set problems that require a different viewpoint from normal in order to be able to solve them.

In my final year at university those of us who had survived the years of student life and overdrafts were required to write a ten thousand word report on a topic within the astronomical field. We were supposed to be serious students by now, and were not allowed such a free rein with our choice of subject matter.

A carefully prepared list was presented, and we chose from that. I selected a curious group of stars known as Wolf-Rayet stars (in honour of their discovers), which are way down at one end of the

stellar life cycle, but which for some peculiar reason are giving off far more energy than they should. Are they nearer than we thought and giving off normal amounts of energy, in which case the standard way of estimating the distances to the stars is proved wrong, or are they really far away, in which case our theories of stellar evolution could do with a spot of revising. Are they artificial beacons in space, manned by beings from other planets? A kind of super-duper Radio One, presumably not playing the same banal drivel that usually occupies our airwaves. Again, send an adventurer there and let him have an explore.

This brings us to the last thing I want to mention in this month's article: the question of life on other planets. Three people in the space of seven days have asked me for my views on this, and personally (and I stress, personally) I think that sheer numbers force me to admit that there must be life, if not intelligent life, elsewhere in our galaxy. Why haven't they visited us, you might ask. Well, we haven't visited them, have we, and we're supposed to be intelligent.

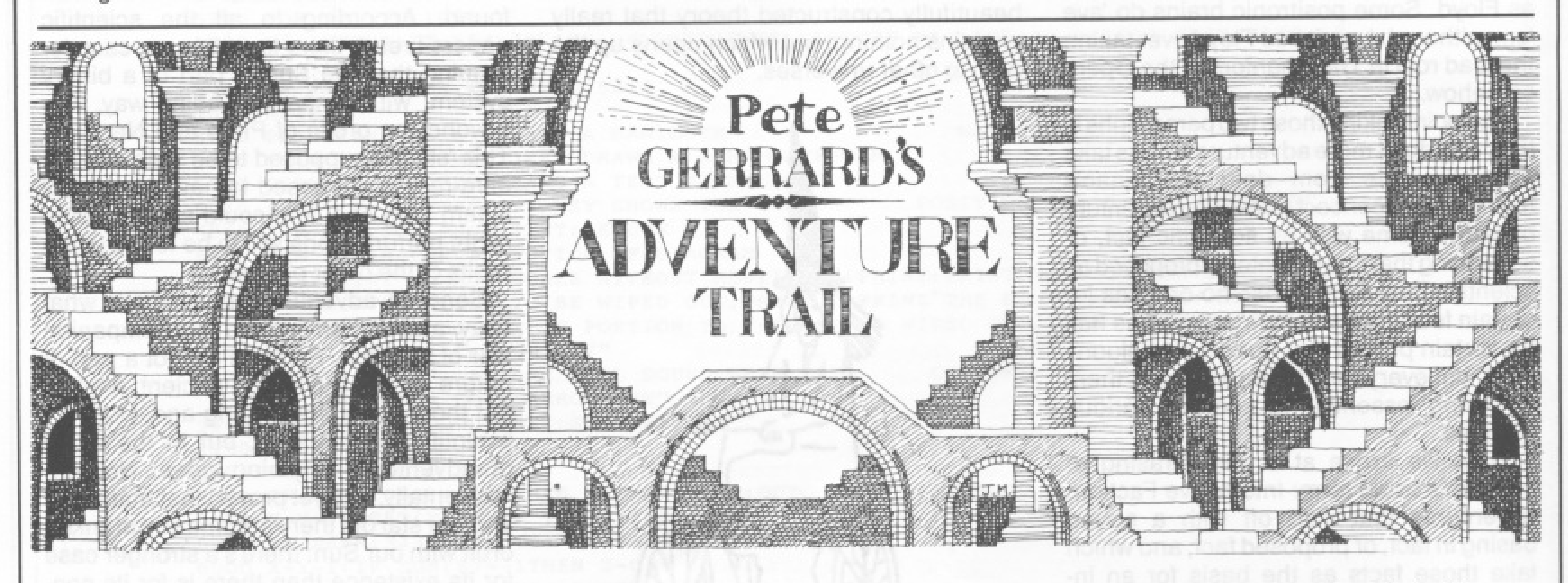
When you look up on a clear night and see the two thousand or so stars that are visible to the naked eye, then when you look through a telescope or a powerful pair of binoculars and see the countless millions more, who can doubt that there are planets orbiting just some of those stars? There are many stars in the same spectral class as our Sun, and presumably they have planets with oxygen-rich atmospheres like ours at a suitable orbit from them, so over the aeons during which our galaxy has existed I would doubt very much that life hasn't appeared somewhere else.

Life everywhere

In the world of science faction (to stick to the term) there seem to be two very different schools of thought about how the universe might proceed. There is the Asimov view, which he sticks to in most of his later stories, that life started out on our planet and spread outwards through the stars, although he does veer from this in some of his earlier works. Not one intelligent lifeform is found on any other planet in all those explorations, which seems a bit naive to me, even if it does give you the opportunity to weave some wonderful stories.

Far more prevalent is the view that life exists all over the place. I must confess that there are times when I find it hard to believe that intelligent life exists on this planet, never mind anywhere else! But, of course, it does. Do UFOs really exist, and if they do then does this indicate the presence of life elsewhere, coming in for a brief look at us before going away again? Why not an adventure from the other side of the coin, where your player is an alien being exploring earth? What would you do if your first sight of earth was a copy of *The Sun* newspaper and a radio playing some mindless zombie disco smash hit record? Turn round and go home?

When you think about it, just one adventure, exploring just one star, and finding just one new race of intelligent beings, could be the start of something big. Look how long *Star Trek* has been a universal (!) favourite. Go for it, adventure writers, hit the libraries, and let's see some intelligent interactive faction appearing on our shelves.



I'll begin with a plea of help, which I pass on from P D Smith in Cardiff and Nick Hodge in Bridgewater. We'll be coming back to Nick later on, don't let him think he's going to escape this lightly. The game in which these noble chaps are stuck is *Return of the Ring*, and on looking through the voluminous files that constitute the Gerrard Answer To Everything we find that these files are sadly lacking as far as this particular game is concerned. Do we have a solution sheet? We do not. Do we have a hint sheet even? Alas, a hint sheet is nowhere to be found.

Do we have the patience and time to sit down and try once again to play the blessed game? We do not, and so on behalf of messrs. Smith and Hodge, to say nothing of myself, can someone somewhere sometime send me a solution? Oh, I just love alliteration. Anyway, a specific problem from Nick Hodge is that he wants to know what to do with the 'units', and also wants to know what he has to do in The Amplifier Room. Haven't a clue, old bean.

Mr. Smith's problems extend even further, and since he puts them down in numerical order I might as well do the same.

1) How do you get the key off the genie?
2) How do you deal with the Trog 8 miles into the forest?
3) What use is the Village of the lost Krells?
4) Where is the healer? I've been to the Temple of Regeneration but there is no-one there.

As Mr. Smith points out, I have not answered any of these in previous issues. And if you're reading this, big brother Mike, he has a sneaky dig at you and says that you haven't either. No wonder you handed the column over ...

So if anyone can help, muchos gracias, as we adventurers say.

Before we get back to the plot, I shall tell you a little tale about Nick Hodge, taken directly from his letter, just to show you what adventuring can be like. And I quote: "After many a happy year playing arcade games, I decided to have a go at a few adventures (don't let the Expert hear about it though). (Don't worry, Nick, your secret is safe with me.) From that fateful day my life has changed dramatically. No longer is it just shooting aliens, but now EXAMINE CHEST, OPEN CHEST and GET TREASURE has entered my vocabulary. A few weeks ago I was content with this, but the inevitable happened. I got stuck. Now it's BASH COMPUTER and HEADBUTT WALL so I write to you for HELP. It might help if I told you all of my problems, but I decided against that. I don't think the world's ready for my problems yet. But here are my adventuring ones:"

End of quote. Problems? You think you've got problems? Have you ever tried to explain to a stone-cold sober policeman (while sitting in the front room of a friends' flat, said friends curled up in hysterics on the sofa), why a drunken parakeet, sozzled on Scotch, is trying to make a nest in your beard? Just one of life's little problems that you must overcome before you can become a true adventure person. Strangely enough, I've never been back to that particular house, I can't begin to think why ...

Back to adventures, and how the old friends love to crop up. In *Trekboer*, writes

Nick, how do you pick up the ice without it melting in your hand? teknalb ni ti yrrac. What's the use of the canteen? If, as I presume, you mean the beaker, then it's used for tenalp eht morf dica gniyrrac. What's the use of the pillow? For no rekaeb eht gnippord. What's the use of the blanket? Semit wef a eno taht derewsna ydaerla ev'I!

In *The Vortex Factor*, after using the white cartridge, Nick finds himself in a dark place. This is not surprising, because dark it is indeed. Reasonably enough Nick wants to stop hitting his head and dying, he would rather like to stay alive. Well, I'm blowed if I'm writing all this lot out backwards, so here goes. Presumably you've not made a candle, which can be done with the aid of some string, and the gruesome instruction melt bird (into wax ball in torch flame), and you can them make and light your candle. If you haven't found the string and the wax bird, then keep trying! You know it makes sense. What is the point of leaving all the treasures in the Curator's Office? Is there ever any logic behind the place chosen by the adventure writer to which all treasures must be returned? Rarely, if ever, so satisfy yourself with the knowledge that this is where the treasure's must go to if you are to get your 100 points and complete the game.

In *Return of the Ring* ... oops ... skates over that one rapidly, until he reaches *Tanglewood*. How do you get the fishing rod off the gnome for the cat? You must send Goliath into the walled garden carrying the wand and cast spell to get the rod. You can then use it to fish for the specs and give them to Foghorn to wear.

In *Syzygy* ... read last month's issue, I refuse to type all that again.

Making money

To take us onto something completely different, we have a letter from one Clark Campbell, in Scotland. Anyone who finishes a letter with Yours adventuringly must be in with a sporting chance of getting mentioned in this column. I shall quote from his letter, because it brings us to a very interesting topic: making money!

And I quote: "I have had a Dragon for over four years. Four years is a long time, though I did write (never did finish) a Basic adventure; it had a simple parser — VERB NOUN type. I had plans for this game: it wasn't your simple 'locations with a number of semi-logical problems' adventure game, but a massive role-playing campaign set in mystic and ancient world 'where magic and monsters abound: comparable to Tolkein at his best!' (*TOLKIEN you son of a three-eared mewlip! He never reads this column ....*) Now where have I heard that line before? I envisaged lots of adventures within adventures — sort of like *Ring of Darkness* — each adventure loading from cassette or disk. Why am I writing all this? I need a little bit of help — could you advise me on anyone who needs a writer for adventure games; not the programming but the actual story line and problems."

End of quote, back to me again. Clark raises something of great interest to any would-be adventure writer with this plea. Perhaps, to begin with, if anyone's interested in doing the programming for Clark's ideas then they could write to him at 33 Shand Street, Wishaw, Scotland ML2 8HN, or indeed if any companies have any enthusiasm for the idea they might also wish to drop him a line.

Novel situation

Getting an adventure game published and marketed is not too far removed from getting a novel released. The situations are fairly comparable, and indeed playing a good adventure should be like reading a good novel. To take the similarity further, just as reputable publishing houses do not go around advertising for novels to be sent to them, so reputable software houses are equally reticent in their own advertising. Everyone welcomes receiving superb books/games, but if you've got any sort of renown then you don't go looking for them, people come to you.

So I can't recommend any company to approach. You know the companies that are currently very big on the adventure front, and those are the ones to go for.

Two or more heads are better than one, when it comes to writing an adventure. As enlightened readers will know, my brother and I have co-written adventures together, with him coming up with the storyline and scenarios and myself doing the actual programming. This worked well, in so far as it went. He is a better writer than wot I am, and I am a better programmer than WOT he is, so we complemented each other nicely.

However, although our efforts were commendable, and I like to think that we produced some above average results, we never achieved earth-shattering greatness, simply because neither of us had (or indeed has) the necessary time to spare in order to produce something truly brilliant. This is why, to get back to Clark's point, I think it is better to submit a storyline to a company that you know from previous efforts are capable of producing the goods. As I'm currently going through the motions of doing just that myself, let's take a look at a typical example — me!

Like a lot of people must do from time to time, I started writing a novel. This was given in two chapter chunks to a friend for proof-reading, and after a while said friend (not the parakeet one!) suggested that it would make a good adventure game. My writing shifted slightly to turn it into the sort of plot that could easily be adapted to the adventure field, and then we both realised that it would make not just a 'good' but a superb adventure game. Writing on the novel stopped, writing on the adventure began.

We have both had adventures published, but soon realised that this was one game where we were not going to be doing the programming. As a programmer, I knew what could and could not be done: this idea could be done, but not by us. As a result of this, we sent a letter to a well-known software house, followed up by a 'phone call, and they asked us to submit a detailed synopsis. Again, as a programmer I know the sort of thing that I would want to see, and so the mammoth task began. This is where I would imagine that so many good adventures do not get off the ground. I was once talking to Pete Austin of Level 9 (he said, name dropping), who astonished me by saying that they quite often get ideas for adventures that run to many pages, all of them hand-written. It does not do your chances any good at all if you are forcing someone to wade through what might be very badly written script. Spelling mistakes and grammatical errors should also be eliminated.

Our adventure has, in the end, some 115 locations. Each one of these has been typed up on a different sheet of paper (and printed out twice, I might add, as my spelling mistakes were removed and additions to the game were made by either myself or my friend), complete with short and long room description, objects that are initially located there, problems to be solved, and possible inputs by the player are also noted down on these sheets.

A separate section of our synopsis has each and every problem (together with the solution to that problem) printed out, cross-referenced back to the big printout of all the locations. A third section has a list of objects and their uses, again cross-referenced back to that big printout. A fourth section contains a detailed solution, a fifth contains the maps for the game, and finally we produced an introductory booklet outlining the characters used in the storyline and the background to the plot of the game.

Impressive

A mammoth task, which has taken quite some time to do, but the end result is an impressive document which (we hope!) will be looked at with far greater respect than a few tatty old sheets of A4 written in leaky biro. And how have we fared? I don't know, we're posting it three days after I've finished this column (one last section to do each, and we're finished). Oh, the agonies of waiting.

So, Mr. Campbell, and anyone else thinking of submitting an idea for an adventure to a large company, send a letter first of all with a brief outline, and if any interest is expressed AT ALL and they ask for a detailed synopsis, you know what you've got to do. And keep a copy of all documents, correspondence, etc. etc. If you think your programming expertise is up to it, then by all means do everything yourself. However, it never does you any harm to admit that you're not capable of doing something. Both of us know that we could never program this game, but we do know that it's a cracking adventure, and between the two of us we've come up with something which is more than twice as good as anything that either of us could do individually.

So, all you non-programmers out there, get writing! See you next month.

Chase the Ace

Gordon Lee plays his cards right

PLAYING cards are the subject of this month's competition problem, plus a bit of magic with a computerised version of *Chase the Ace*.

The effect of this trick as as follows: four cards are displayed on the screen, an ace and three face down cards. The ace can be switched with either of its neighbours by using the keyboard. A volunteer is then invited to repeat this switching operation as often as he wishes while your back is turned. When he has finished you specify a card at a certain position and instruct him to remove it from the display by pressing an appropriate key. He is then asked to switch the ace one further time before you name the positions of two of the remaining cards to be eliminated. The one card left displayed on screen is the ace, despite the fact that your back had been turned throughout the whole operation land you would have no way of knowing the sequence of switches that were being made.

When the listing shown is run the cards are drawn on screen. To switch the position of the ace with one of its neighbouring cards press the key corresponding to the number below the face-down card to be moved. When the ace is at the end of the row it can be exchanged only with one other card, but when it is at one of the central positions there is a choice of two switches which can be made. After a key has been pressed the scrfeen display is updated and a tone sounds to indicate that another switch has been made. The method of switching the cards should be demonstrated to the volunteer. "Don't press a key until you hear the tone", you must say, "or the computer won't be ready!".

Now, turn your back to the computer and let him switch the cards around as often as he wishes. When he indicates that he is finished you tell him to eliminate one of the end cards by pressing either key L or R to denote the left or the right hand end. You, of course, specify which of these cards is to be removed. Finally, ask him to make one more switch before specifying a further two cards to be eliminated. The remaining card proves to be the ace!

The question is, how do you know which cards to name, as you have no way of knowing the order in which the cards are being switched? The secret is in the tone which is sounded to indicate that the next key is to be pressed. After demonstrating to the volunteer how to switch cards, remember where the ace is before turning your back. It is not necessary to remember the actual position, only whether it is an odd or even

```
  10 DIM A(60),B(60)
  20 A=2:Y=32:L=36:R=186
  30 GOSUB 1000
  40 GOSUB 2000:GOSUB 3000
  50 GOTO 40
1000 PMODE4:SCREEN1,1:PCLS0
1010 FOR X=30 TO 180 STEP 50
1020 LINE(X,Y)-(X+40,Y+64),PSET,BF:NEXT
1030 LINE(50,110)-(50,122),PSET
1040 CIRCLE(100,113),4,1,1,0.6,0.2:LINE(100,117)-(97,122),PSET:L
     INE(97,122)-(105,122),PSET
1050 CIRCLE(150,113),4,1,1,0.6,0.3:CIRCLE(150,119),4,1,1,0.8,0.4
1060 DRAW"BM200,122;U12G6R10"
1070 FOR F=36 TO 64 STEP4:LINE(F,36)-(F,92),PRESET:NEXT
1080 FOR F=36 TO 92 STEP4:LINE(36,F)-(64,F),PRESET:NEXT
1090 CIRCLE(100,52),6,0:CIRCLE(94,62),6,0:CIRCLE(106,62),6,0
1100 PAINT(100,52),0,0:PAINT(94,62),0,0:PAINT(106,62),0,0
1110 LINE(100,65)-(96,75),PRESET:LINE(96,75)-(104,75),PRESET:LIN
     E(104,75)-(100,65),PRESET
1120 PAINT(100,70),0,0
1130 GET(36,36)-(64,92),B,G:GET(86,36)-(114,92),A,G
1140 RETURN
2000 FOR C=L TO R STEP 50:Z=(A-1)*50+36
2010 IF C=Z THEN PUT(C,36)=(C+28,92),A,PSET
2020 IF C<>Z THEN PUT(C,36)=(C+28,92),B,PSET
2030 NEXT
2040 SOUND 200,4
2050 RETURN
3000 A$=INKEY$:IF A$=""THEN 3000
3010 IF A$="L"OR A$="R"THEN 3050
3020 IF A$<"1"OR A$>"4"THEN 3000
3030 V=VAL(A$):IF ABS(V-A)=1 AND V<>J THEN 3040 ELSE 3000
3040 A=V:GOTO 3080
3050 IF A$="L"THEN X=L:L=L+50:J=1 ELSE X=R:R=R-50:J=4
3060 LINE(X-6,30)-(X+34,130),PRESET,BF
3070 IF L=R THEN PLAY"02L16CDEFGABAGFEDC"
3080 RETURN
```

Prize

SHUFFLING towards a reply to Gordon's cutting question could turn up another lucky card — a discount card, or even a copy of **Pyradventure** or **Underbeings of Croth** from **Dragonfire Services**, whose quest for the ultimate program under a fiver continues unabated. The pick of their list can be yours if you can find the card marked X . . . .

Rules

When you have followed the clues to the spot marked X and dug up a solution, bury it again deep within an envelope marked APRIL COMPETITION (what d'you mean, which year??) along with a printout and any comments you wish to append, and send it off to us at the usual place. Don't forget to include your name and address, preferably not hidden in the program listings!

Now for the tiebreaker: literature time! (Actually, the mind of mathematician Lewis Carol was more cunningly convoluted, if 'tis possible, than even that of Gordon Lee.) "Why is everyone running around like headless chickens in a barnstorm?" asked Alice, surveying the *Dragon User* office. What was the Red Queen's reply?

January winners

Close, this. Lots of entries, no wrong answers, lots' of nice neat programs, tiebreakers high and low . . . after much sweat, the winners are: Don Robertson of Epsom, Paul Priestland of Lechlade (who won the Crossword as well, jammy fellow), Paul Weedon of Wotton-under-edge (get a new ribbon, Paul, please??), Eric Hastead of Erith, Austan Henderson of Bromsgrove. The First five all get the adventure *Larkspur Waldorf is Trapped*, kindly supplied by John and Helen Penn and the author John Smallwood. The rest to get discount vouchers from the Penns: John Mackin of Troon, Rochard Long of Camberley, Keith David of Crawley, Phil Sapiro of Liverpool, S A Siddiqui of Chiswick, E A Newman of Addlestone, F J Taylor of Acklam, Les Simpson of Littleport, Denis O'Mulloy of Comberton, R Hames of Milton Keynes, D J Gray of Cleveland, Ian Huggins of Caerphilly, and Alan Thomas of Staplehurst. L W Turner is out of the top twenty, but requested a special mention "just to show my family that I'm not past it", and him only 17, too.

In answer to somebody's question, anyone *can* be excused the tiebreaker, but in the event of a tie, they won't have a tiebreaker, will they? We aren't looking for genius, just evidence that you can still think in English after a hard week's programming. And *someone* has to make the editor laugh once a month.

This month's best tiebreaker from Keith David: "No L, No L, No L, No L, Is there somebody out there with a keyboard to sell?" No prizes for guessing which carol that's based on . .

Solution

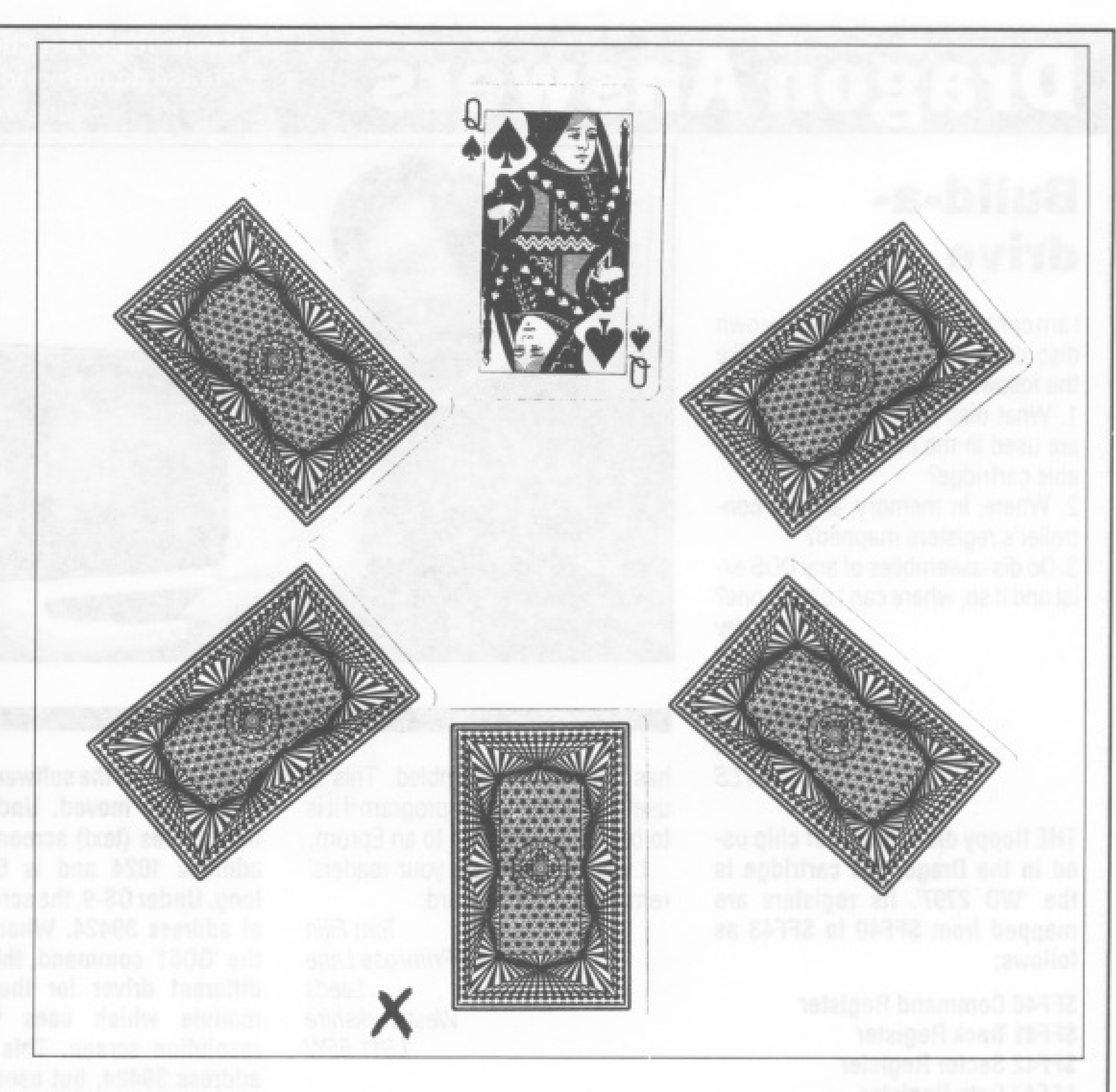
This month's solution is opposite.

Footnote

Apologies to those who have not received *Chuckie Egg/Screaming Abdabs*. Computape are moving house at the moment, and doubtless all will be restored when they have a new roof over their heads.

numbered card. Now, each time that the tone sounds the ace will change from an even to an odd position or vice versa. So, as the switches are being made simply count 'odd, even, odd, even ...' and so on. When this is done you will not know the absolute position of the card, but you will know if it is in an odd or an even numbered position. Of it is odd then you can safely instruct that key R can be pressed to remove the right hand card. Similarly, if the ace ends up as 'even', then the left hand card should be named. When you ask for one final switch to be made, this places the ace in the central position of the three remaining cards, so, after a great show of concentration, eliminate the left and then the right hand cards, the ace will be left behind, and the trick can be brought successfully to its conclusion.

For a more subtle presentation of the trick the sound instruction can be removed from the program listing, but if this is done you must be sure of being able to hear the actual keys as they are being pressed.



Competition

Shown here are six cards taken from a standard pack. They have been dealt as shown and one of the cards has been turned face up. Can you determine the values of each of the cards from the following information (Ace = 1, Jack = 11, Queen = 12 and King = 13):

The sum of the two cards to the right of the queen is equal to the sum of the two cards to the left of the queen.

The queen plus the two cards either side of her sum to the same total as card X plus its two neighbours.

The total value of all six cards is 52 (the number of cards in a pack).

There is one pair of cards of equal value (which may or may not include the queen) — and one pair only.

There is a 9 among the cards but it is not the card one place clockwise from X.

If I were to tell you the value of card X you would be able to determine the values of each of the cards!

.... but can you do it without this information?

The Answer

This is Gordon Lee's own solution to the January competition
see page 22 for results

ANSWER: A=2697 (A^2=7273809) and B=1489 (B^2=2217121)

SOLUTION: The two values for A and B must both be in the range of 1000 to 3162 as they both have seven digit squares. If both values were generated independently, there would be over four and a half million permutations possible, so much unnecessary work is eliminated by first selecting only those values possible for 'A' before considering 'B' at all. For example, we know that the last digit of A must interlock with the first digit of A squared. Also, A squared must have a 7 as its third digit, and an 8 as its fifth.

In the listing variable S1 is the square of A, and the string variables S1$ and A$ are their string equivalents. Because of the 'ghost' character placed by the Dragon at the front of any string variable created by the STR$ command, the second instruction in lines 110 and 130 remove this.

This makes the programming much more logical as the first character in the string is now the first digit in the number, and so on.

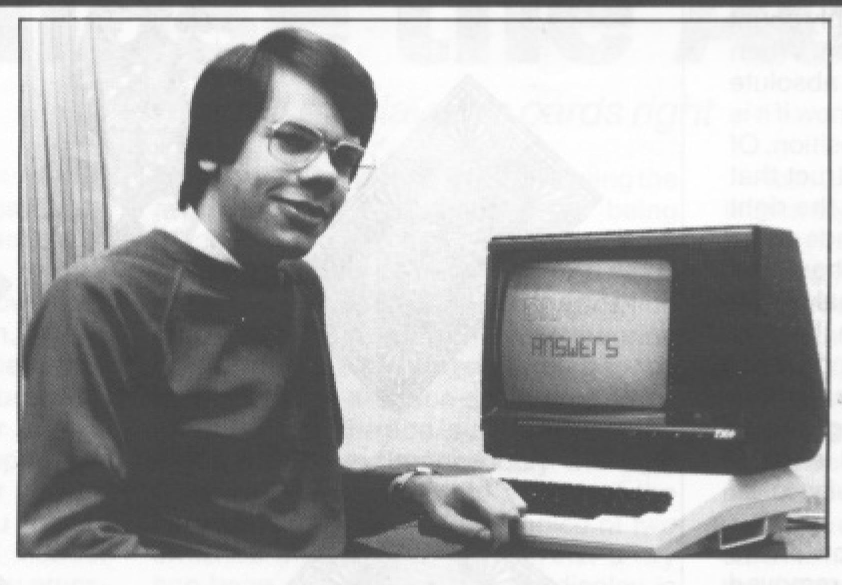
Lines 140 to 160 test each string to check that interlocking digits correspond, and only once a possible value for A has been found does the program check for any 'B' values. This is done in the same way as with the 'A' variables, except that S2 and S2$ are the corresponding variables holding the value of B squared.

Only when all eight interlocking digits pass the test is the result printed out at line 260.

```
100 FOR A=1000 TO 3162
110 A$=STR$(A):A$=MID$(A$,2)
120 S1=A*A
130 S1$=STR$(S1):S1$=MID$(S1$,2)
140 IF MID$(A$,4,1)<>MID$(S1$,1,1) THEN 280
150 IF MID$(S1$,3,1)<>"7" THEN 280
160 IF MID$(S1$,5,1)<>"8" THEN 280
170 FOR B=1000 TO 3162
180 B$=STR$(B):B$=MID$(B$,2)
190 IF MID$(B$,4,1)<>MID$(S1$,7,1) THEN 270
200 S2=B*B
210 S2$=STR$(S2):S2$=MID$(S2$,2)
220 IF MID$(B$,1,1)<>MID$(S2$,7,1) THEN 270
230 IF MID$(A$,1,1)<>MID$(S2$,1,1) THEN 270
240 IF MID$(S2$,3,1)<>"1" THEN 270
250 IF MID$(S2$,5,1)<>"1" THEN 270
260 PRINT A$;" ";S1$;" ";B$;" ";S2$
270 NEXT B
280 NEXT A
```

Dragon Answers

If you've got a technical question write to Brian Cadge. Please do not send a SAE as Brian cannot guarantee to answer individual inquiries.



Build-a-drive

I am contemplating building my own disc drive interface and would like the following information:

1. What disc drive controller chips are used in the commercially available cartridge?
2. Where, in memory, are the controller's registers mapped?
3. Do dis-assemblies of any DOS exist and if so, where can I obtain one?

P M Buckley
72b High Street
Whitton
Twickenham
Middx
TW2 7LS

THE floppy disc controller chip used in the Dragondos cartridge is the 'WD 2797'. Its registers are mapped from $FF40 to $FF43 as follows;

$FF40 Command Register
$FF41 Track Register
$FF42 Sector Register
$FF43 Data Register

The disc motor control latch is mapped at address $FF48. This controls the disc motors for all four drives.

To my knowledge, there are no commercially available listings of any of the Dragon DOSs. In any event, these would of course be copyrighted. You will need a lot of in depth knowledge of both the Dragon and Disc Interfacing to design your own cartridge — it took Dragon Data six months to get Dragondos finally working correctly.

RAM card running

I read with interest the letter from Phil Callaghan (Jan 88) headed *ROM can't be RAM*. I have in fact been running such a RAM card, which I designed, for the past 16 months on my Dragon.

The reason why I have not produced the card commercially is the possible lack of sales of such an item. The full CART area can be used (less 256 bytes 10 functions). The cost of the card, to fit into an old Dragon Data cartridge housing, would be approximately £27 for an 8K RAM and £34 for 16K.

My card has a 'write protect switch' which can produce ROM only functioning after the program has been fully assembled. This is useful for testing your program if it is to be finally blown on to an Eprom.

I would appreciate your readers' remarks on such a card.

Tom Filin
13 Primrose Lane
Leeds
West Yorkshire
LS11 5EW

I am still getting quite a few letters about RAM cartridges. If you are interested in this particular design then take the trouble to contact Tom at the address given.

Screens resolved

I have tried to write a program which shifts to the low resolution screen, after using *Stylograph* and OS-9. The program listed below crashes when using OS-9, but not with normal Basic. Where are the low resolution screens stored in OS-9?

```
pshs a,b,x
ldx #$ffc8
sta 4,x
sta 2,x
sta 1,x
sta -2,x
sta -4,x
sta -6,x
sta -8,x
lda $ff22
anda #$07
sta $ff22
puls a,b,x,pc
```

Geir Hovland
Legdavegen
6943 Naustdal
Norway

THE program you are using tells the hardware to display the screen from a different memory location. it does not tell the software that the screen has moved. Under Basic, the low res (text) screen starts at address 1024 and is 512 bytes long. Under OS-9, the screen starts at address 39424. When you run the 'GO51' command, this loads a different driver for the KBVDIO module which uses the high resolution screen. This starts at address 39424, but uses up 6144 bytes rather than 512. Unfortunately, there is no 'GO32' commands supplied with OS-9 and it is no simple task to re-load the low resolution screen driver. The simple answer is to power off and on to release the extra memory and return to the low res screen.

DLOAD what?

WHILE peeking through the Basic ROM I came across the list of commands and functions. Amongst these was one called 'DLOAD'. This is not in my manual, can you tell me what is is supposed to do? All I get is ?IO ERROR.

Pete Donaldson
Banbury

THE DLOAD (and DLOADM) command is the serial port equivalent of the CLOAD (and CLOADM) command. The code was partly written in the Dragon 32 ROM, but then abandoned and the command simply causes an IO Error.

The Dragon 64 has a built-in RS232 port and so the code is complete in the '64 ROM (it uses channel -3 which is the internal RS232 port channel). The format of the data expected t o come in at the port is the same as the format for cassette data (which is too complex to go into here). As far as I know, no commercial software has ever used the command as the format is so complicated that the source computer (connected to the RS232 port) could not easily produce the data in the required format.

The command is really just a 'dinosaur' from earlier versions of Microsoft's 6809 Basic.

Where is my 32K?

JUST recently I have had my Dragon 32 upgraded to 64K but now I need to know how to access the extra 32K. All that appears on the screen when the computer is switched on and I type ?MEM is 24871.

I have also just recently purchased a Cumana single disc drive with Dragondos/Superdos E5. I would like to know how to use the system to the full so could you tell me what sort of disc I need. Am I right in thinking that I need a Dragondos disc and, if so, do you know where I can get hold of one?

G J Waddilove
47 Buckshaw Hall Close
Astley Village
Chorley
Lancs
PR7 1SX

UNLESS you have a Dragon 64, it not possible to access the extra 32K RAM from Basic as Basic itself sits in ROM which overlays the extra RAM. With a '64, typing EXEC immediately after power up will copy and switch out the ROM chip. It is very unlikely that your upgrade included upgrading the Basic ROMs as well.

However, if you have a disc interface plugged in then it is not possible to access this extra RAM (even on a Dragon 64) as the disc ROM also overlays the extended RAM and has no built-in provisions for copying itself to RAM.

The extra RAM may be used by (a few) commercial programs (eg adenture games, word processors etc), but to access it yourself you'll need to resort to machine code.

The type of disc you need for your drive is any 5.25in double density disc. Don't worry about it being single or double sided on the label — only one side will be used anyway. You'll need to use the DSKINIT command to format the disc before using it. If you want to use someone else's disc then this must also have been formatted as a Dragondos disc.